GE Fanuc Automation

Programmable Control Products

CIMPLICITY® Control: Using the Sequential Function Chart Editor

GFK-1385 *November 1997*

Warnings, Cautions, and Notes as Used in this Publication

Warning

Warning notices are used in this publication to emphasize that hazardous voltages, currents, temperatures, or other conditions that could cause personal injury exist in this equipment or may be associated with its use.

In situations where inattention could cause either personal injury or damage to equipment, a Warning notice is used.

Caution

Caution notices are used where equipment might be damaged if care is not taken.

Note

Notes merely call attention to information that is especially significant to understanding and operating the equipment.

This document is based on information available at the time of its publication. While efforts have been made to be accurate, the information contained herein does not purport to cover all details or variations in hardware or software, nor to provide for every possible contingency in connection with installation, operation, or maintenance. Features may be described herein which are not present in all hardware and software systems. GE Fanuc Automation assumes no obligation of notice to holders of this document with respect to changes subsequently made.

GE Fanuc Automation makes no representation or warranty, expressed, implied, or statutory with respect to, and assumes no responsibility for the accuracy, completeness, sufficiency, or usefulness of the information contained herein. No warranties of merchantability or fitness for purpose shall apply.

The following are trademarks of GE Fanuc Automation North America, Inc.

Alarm Master	GEnet	PowerMotion	Series One
CIMPLICITY	Genius	ProLoop	Series Six
CIMPLICITY PowerTRAC	Genius PowerTRAC	PROMACRO	Series Three
CIMPLICITY 90–ADS	Helpmate	Series Five	VuMaster
CIMSTAR	Logicmaster	Series 90	Workmaster
Field Control	Modelmaster		

All rights reserved.
©1997, GE Fanuc Automation North America, Inc. (Charlottesville, USA) and Cegelec (Levallois-Perret, France)

.

Content of This Manual

This manual describes the features that are used to program Series 90™-70 and Series 90-30 PLCs using the CIMPLICITY Control SFC Editor, Release 2.0. This manual does not describe the features and functions that CIMPLICITY Control software shares with the SFC Editor. For information on these features and functions, please refer to *Using CIMPLICITY® Control*, GFK-1295.

Chapter 1. **Introduction.** Provides a brief overview of the SFC editor.

Chapter 2. **Understanding SFC Programming.** Provides a general introduction to SFC programming, including basic structures and a sample application.

Chapter 3. **Working in the SFC Editor.** Describes how to use the CIMPLICITY Control SFC editor software to create and edit SFC logic.

Chapter 4. **Troubleshooting.** Describes how to use the diagnostic tools available in the SFC editor.

Chapter 5. **Printing SFC Documents.** Describes how to print your SFC program.

Appendix A. **Common User Errors.** Describes error conditions that are unique to sequential function chart logic blocks and the ladder diagram logic within them.

Appendix B. **PLC Limits.** Contains a listing of SFC performance parameters for target PLCs.

Appendix C. **Glossary.** Provides definitions of terms pertaining to SFC programming.

Related Publications

GFK-1295	*Using CIMPLICITY® Control*
GFK-1296	*Series 90™ Systems Reference Manual*
GFK-1192	*Series 90™-70 System Manual*
GFK-0262	*Series 90™-70 Programmable Controller Installation Manual*
GFK-0646	*C Programmer's Toolkit for the Series 90 PLCs User's Manual*
GFK-0600	*Series 90™-70 Programmable Controller Data Sheets*
GFK-0356	*Series 90™-30 Programmable Controller Installation Manual*

At GE Fanuc Automation, we strive to produce quality technical documentation. After you have used this manual, please take a few moments to complete and return the Reader's Comment Card located on the next page.

Libby Allen
Senior Technical Writer

Contents

Chapter 1 Introduction 1-1

SFC Defined 1-2
What You Will Need 1-2
Installing the Software 1-2
Using Online Documentation 1-2

Chapter 2 Understanding SFC Programming 2-1

RLD and SFC 2-1
Steps 2-3
    Step Action Logic 2-3
Transitions 2-4
Example SFC Control Sequence 2-5
Series 90 PLC Scan Cycle 2-6
SFC Evolution 2-8
    Step Status Values 2-8
    SFC Evolution Example 2-9
Basic Control Structures 2-11
    Simple Sequence 2-11
    Divergence of a Selective Sequence 2-12
    Convergence of a Selective Sequence 2-13
    Divergence of a Simultaneous Sequence 2-14
    Convergence of a Simultaneous Sequence 2-15
    Sequence Skip 2-16
    Sequence Loop 2-17
    Jump 2-18
    Cycle 2-19
Rules for Control Structures 2-20
    IEC Compliance 2-23
SFC Reset Function 2-25

Chapter 3 Working in the SFC Editor 3-1

Getting Started 3-2
    Creating a Main SFC Block 3-2
    Creating a New SFC Block 3-3
    Editing an Existing SFC Block 3-3
    Importing a Logicmaster 90 SFC Program 3-3
SFC Editor Windows 3-4

Topology Window ... 3-6
How Elements are Displayed ... 3-6
SFC Element Names ... 3-6
Step-Action Association ... 3-7
Comments ... 3-7
Logic ... 3-7
Animation of SFC Topology ... 3-7
SFC Toolbar ... 3-8
SFC Status Bar ... 3-8
SFC Pop-up Menu ... 3-9
Key Functions ... 3-10
Variable Declaration Editor Window ... 3-12
Terminal Language Window ... 3-12
Setting Edit and Display Features ... 3-13
Changing SFC Editor Options ... 3-13
General ... 3-13
Action ... 3-14
Transition ... 3-15
Using Lock Cursor ... 3-16
Customizing the Topology Window Display ... 3-16
Grid Display Options ... 3-16
Zoom ... 3-16
Display Mode ... 3-17
Viewing SFC Limits ... 3-18
Adding an SFC Network to the Block ... 3-18
Building an SFC Network ... 3-19
Creating Steps and Transitions ... 3-19
Steps ... 3-20
Transitions ... 3-21
Creating Actions ... 3-21
Connecting SFC Elements ... 3-23
Connection Band ... 3-23
Disconnection Band ... 3-23
Connector ... 3-24
Directed Link ... 3-24
Rules for Connecting Steps and Transitions ... 3-25
Editing an SFC Network ... 3-26
Navigating in the SFC Editor ... 3-26
Going to an SFC Network ... 3-26
Searching for SFC Elements ... 3-26
Selecting Elements ... 3-30

Moving Elements ... 3-30
Disconnecting SFC Elements ... 3-30
Cutting SFC Elements ... 3-30
Copying SFC Elements ... 3-31
Pasting SFC Elements ... 3-31
Paste ... 3-31
Paste (with Logic) ... 3-32
Deleting SFC Elements ... 3-33
Undoing a Delete ... 3-33
Viewing and Editing SFC Element Properties ... 3-33
Editing Transition Logic ... 3-34
Editing Action Logic ... 3-34
Editing the Action Pool ... 3-35
Editing Step-Action Associations ... 3-37
Working with Preprocessing and Postprocessing Logic ... 3-39
Creating Preprocessing/Postprocessing Logic ... 3-39
Editing Preprocessing/Postprocessing Logic ... 3-39
Clearing Preprocessing/Postprocessing Logic ... 3-40
Documenting Your SFC Project ... 3-40
Comments ... 3-40
Description ... 3-41
Summary ... 3-41
Storing Your SFC Program to the PLC ... 3-41
Notes for Logicmaster 90 Users ... 3-42
IEC-Compliant Structures ... 3-42
Automatic Creation of Branches ... 3-42
Grid Organization ... 3-42
Drag-and-Drop Editing ... 3-42
Selection of Multiple Elements ... 3-43
Directed Links ... 3-43
Step Timer Updates ... 3-43
Compatibility ... 3-43

Chapter 4 Troubleshooting ... 4-1

Analyzing SFC Networks ... 4-2
Setting Time Parameters ... 4-2
Table ... 4-4
Monitoring Transition Conditions ... 4-6

Forcing Transitions 4-9
Toggle Transition 4-9
Override Transition 4-9
Recording and Viewing Evolution History 4-10
Table 4-11
Buttons 4-11

Chapter 5 Printing SFC Documents 5-1

Topology Options 5-2
Actions 5-3
Transitions 5-3

Appendix A Common User Errors A-1

General Errors A-2
Transition Logic Errors A-2
Other Logic Errors A-3
General SFC Element Errors A-3
SFC Top Level Errors A-4

Appendix B PLC Limits B-1

Appendix C Glossary C-1

Figure 2-1. PLC Scan Cycle 2-7
Figure 2-2. Convergence of Simultaneous Sequences..... 2-21
Figure 2-3. Nesting of Dissimilar Sequences 2-22
Figure 2-4. Non-IEC-Compliant Convergence..... 2-23
Figure 2-5. Non-IEC-Compliant Divergence 2-24
Figure 3-1. SFC Editor Windows..... 3-5
Figure 3-2. Directed Link Example 3-25

Contents

Table 2-1. Sequence of Events for the SFC Evolution Example 2-10
Table 3-1. SFC Pop-up Menu Choices 3-9
Table 3-2. Keyboard Shortcuts 3-10
Table B-1. Target PLC Constraints for Series 90-70 PLCs B-2
Table B-2. Target PLC Constraints for Series 90-30 PLCs B-3

Chapter 1 Introduction

This manual explains how to use Sequential Function Chart (SFC) language with CIMPLICITY Control software to program your logic and create an application program for your programmable controller.

This manual does not describe the features and functions of CIMPLICITY Control software that are shared with the SFC Editor. For information on these features and functions, please refer to *Using CIMPLICITY® Control*, GFK-1295. The *Series 90™ Systems Reference Manual*, GFK-1296, provides more detailed information about PLC systems. For information on programming instructions and timing information pertaining to the 90-70 software, refer to the *Series 90™-70 System Manual*, GFK-1192.

The CIMPLICITY Control SFC Editor, release 2.0 supports the following PLCs.

Series 90-30 PLCs	**Series 90-70 PLCs**
CPU 361	CPU 731
CPU 360	CPU 771
CPU 352	CPU 772
CPU 351	CPU 781
	CPU 782
	CPM 914, 915
	CPM 924, 925

SFC Defined

SFC is a graphical, state language that is IEC-compliant. It was specifically developed for controlling sequential processes and offers a graphical representation of the functions of a sequential automated system as a series of steps and transitions.

For a better understanding of what SFC does and the relationship between SFC and ladder diagram (LD) programming, refer to Chapter 2.

What You Will Need

Chapter 1 of *Using CIMPLICITY Control,* GFK-1295, describes the requirements for installing and running CIMPLICITY Control software. You should also refer to the Important Product Information document shipped with your release for any last-minute changes or special operation notes about the installation procedure.

Installing the Software

Before you can begin using the features and functions described in this manual, you must first install the software. If you have not yet installed the CIMPLICITY Control software, follow the installation procedure described in Chapter 1 of *Using CIMPLICITY Control,* GFK-1295.

Any options you have purchased, such as the SFC Language Editor, are automatically installed when you install CIMPLICITY Control. For information on available options, contact your GE Fanuc distributor.

Using Online Documentation

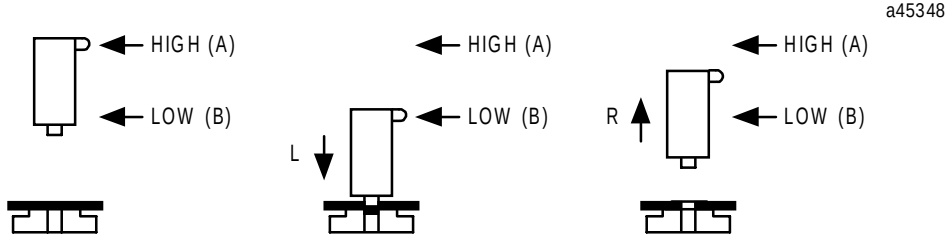
CIMPLICITY Control software includes detailed Help screens that are loaded onto the hard disk of your programmer during the software installation procedure and are readily accessible. To see a Help screen specific to the active window or dialog box, press F1.

Chapter 2

Understanding SFC Programming

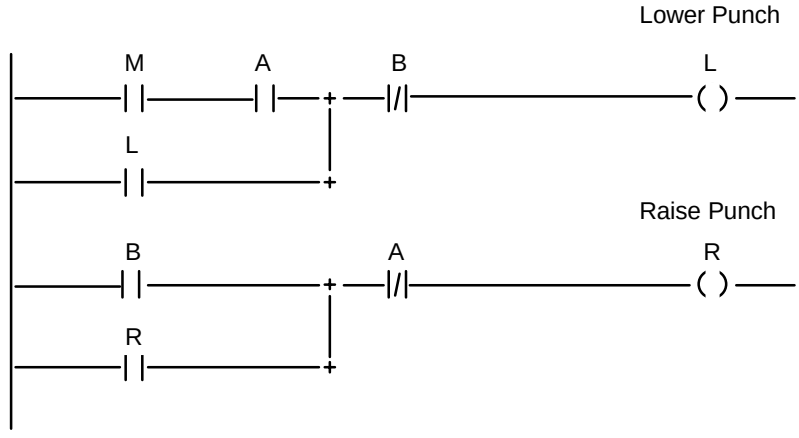
RLD and SFC

The traditional Relay Ladder Diagram (RLD) logic (also called Ladder Logic) has been widely used for designing and representing logic control systems since the relay was invented. For combinational logic control systems, where outputs or actions are directly dependent on the states of inputs or conditions, RLD is excellent. However, for control problems where actions are sequential or time dependent, RLD logic alone can become cumbersome. Consider the simplified example of a semi-automatic punch.



In this example, a semi-automatic punch is initially in the raised position (A). When the operator presses the Start pushbutton, the punch descends, pierces the metal plate at position (B), and completes the cycle by returning to the raised position.

In the ladder diagram approach, you have to take more into account than the fact that pressing the Start pushbutton initiates the movement of the punch. You must also consider that the operator will release the pushbutton and that he could press the pushbutton before or after the punch is in the rest position. This additional information has no bearing on the movement of the punch and complicates the description of the function.



What is needed is a method of representing sequential control problems in a sequential manner, showing the various actions to be taken in a step and indicating the conditions that must be fulfilled before advancing to the next step. To observe some of the differences between SFC and RLD, compare the diagram above to the one on page 2-4.

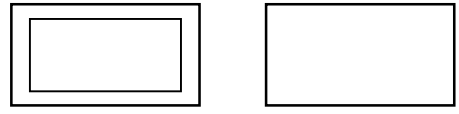
SFC is a language specifically developed for programming industrial sequential control systems. The SFC Editor represents the functions of a sequential automated system as a series of steps and transitions. Each step represents commands or actions that are either active or inactive. The flow of control passes from one step to the next through a conditional transition that is either true (1) or false (0). If the transition logic evaluates true (1), control passes from the current step (which becomes inactive) to the next step, which then becomes active.

Each control function can be represented by a group of steps and transitions, called a Sequential Function Chart (SFC).

Steps

Two types of steps can be used in an SFC: *initial steps* and *regular steps*. The initial step is executed either the first time the SFC block is executed or as a result of the SFC_RESET function, described on page 2-25. There can be *only one* initial step per SFC network. The initial step cannot appear within a simultaneous structure, but can appear anywhere else.

A regular step is executed if the preceding transitional logic in the SFC network makes the step active. There can be one or many regular steps in an SFC network. Their display in the SFC topology looks as follows:



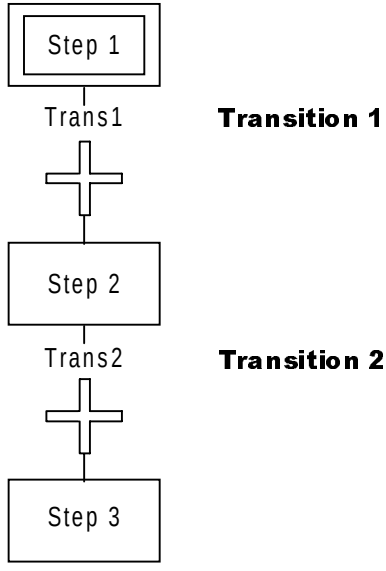
Initial Step Step

Step Action Logic

Action logic, the logic associated with a step, is executed when the step is active. An action consists of zero or more rungs programmed in a terminal language, such as Ladder Diagram. A step can have more than one action associated with it (depending on the limitations of the target PLC) and an action can be associated with more than one step.

Transitions

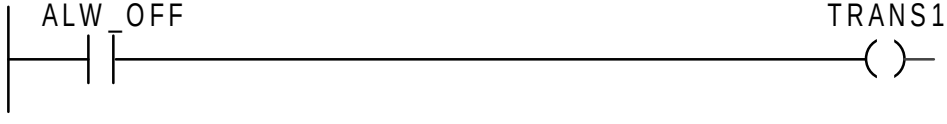
Transition logic is programmed in a terminal language, such as Ladder Diagram. A transition can contain only one executable rung plus comment rungs and is displayed in the topology as shown below.



By default, the SFC editor automatically creates a logic rung for each transition when it is created (unless you have turned this option off in the SFC Editor Options dialog box). Note that the ALW_OFF (always off) contact is used in the default rung for safety reasons. You must edit this rung to meet the needs of your application. The following rung sets the variable TRANS1 for transition Trans1.

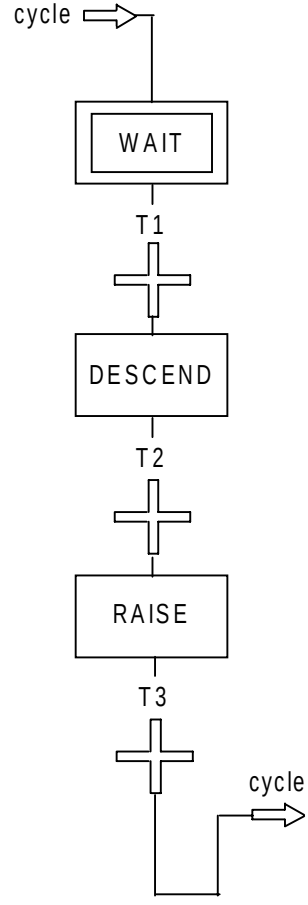
Note

As defined by IEC, a transition cannot have any effect other than setting the transition itself.



Example SFC Control Sequence

A semi-automatic punch starts in the raised position. When the operator presses the Start pushbutton, the punch descends, performs the punching operation, and completes the cycle by returning to the raised position. The SFC sequence for this example is shown in the following illustration.



In the SFC sequence shown in the illustration, the initial step (WAIT) contains action logic that is performed the first time the SFC network is executed and each time the transition T3 goes active causing the topology to cycle. (The initial step will also become active if the SFC is reset via the SFC_RESET function block.) A regular step is represented by a square which can also contain a name (such as DESCEND or RAISE) identifying the principle function of that step. Action logic associated with each step and transition (T1, T2, and T3) is programmed in the Terminal Language window of the SFC Editor.

Series 90 PLC Scan Cycle

The PLC scan continuously executes the MAIN block and any called SFC block when the PLC is in RUN mode. Each SFC block includes preprocessing (optional), sequential processing, and postprocessing (optional) logic. The scan cycle is illustrated in Figure 2-1.

Preprocessing logic, which is optional, must be programmed in a terminal language, such as Ladder Diagram. If preprocessing logic is used in a main block, it is processed at the start of every scan. If it is used in a sub-block, it is processed at the beginning of each executed call to the block. The amount of logic that can be programmed is limited only by the size of the program block.

Preprocessing logic can be used to process events that affect the sequential processing section of the program. These events could include:

- Initialization
- Operator commands
- Resetting the SFC to the initial state

Sequential processing consists of evolving the SFC to its next state and processing the action logic of any steps that become active. Only the logic associated with active steps and transitions is scanned by the PLC, leading to a significant reduction of scan time. For more information, see "SFC Evolution."

Postprocessing logic, which is optional, must be programmed in a terminal language, such as Ladder Diagram. It is executed each time the program block is called, after the evolution of the sequential function charts. That is, after all action logic is executed, but before the program block is exited. The amount of logic that can be programmed is limited only by the size of the program block.

Postprocessing logic could be used to process safety interlocks, for example.

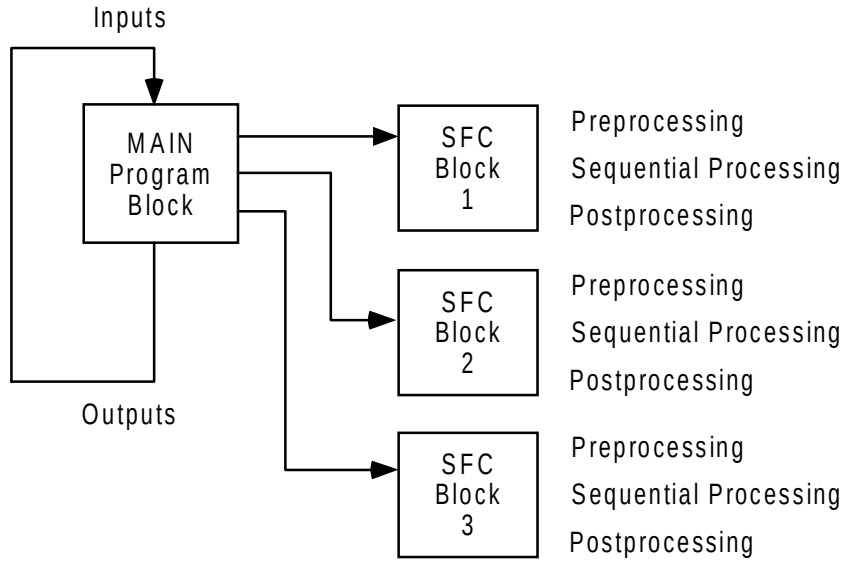


Figure 2-1. PLC Scan Cycle

SFC Evolution

A step can be either active or inactive. A transition is active (evaluated) only when the step immediately preceding it is active. If a transition evaluates true (ON), the software makes the immediately preceding step(s) inactive and the immediately succeeding step(s) active. This process is referred to as *evolving the SFC*. The sequence of operations for any given sequential processing portion of the PLC scan consists of:

1. Evaluating the transitions following all active steps.
2. Evolving the SFC as needed.
3. Executing the action logic associated with the active steps.

The process of evolving the sequential function chart is animated in the Topology window. Active steps and transitions are displayed with backlighting when the PLC is online.

For a discussion of the logic execution flow and SFC evolution of a sample SFC segment, see page 2-9.

Step Status Values

The SFC Editor maintains three local-scope, read-only status values for each step:

***stepname*.x** (step action bit) Set when all action logic within the step has been executed. SFC_RESET function sets this bit to OFF (0). Data type is BOOL.

***stepname*.t** (step timer word) Displays the number of units of time that the step has been active, relative to the time base. SFC_RESET function initializes this word with 0. Data type is WORD.

***stepname*.f** (step fault bit) Set when the step does not attain minimum time or when it exceeds maximum time. SFC_RESET function sets to OFF (0). When the programmer is online and the program is running, a step that has a time fault is identified by red backlighting. Data type is BOOL.

To set the time base for the current SFC block and time limits for individual steps, refer to “Setting Time Parameters.”

SFC Evolution Example

The logic execution flow is described below for a sample SFC segment to illustrate how transitions and states become active in SFC evolution. Table 2-1 summarizes the sequence of events.

- When an SFC block is initially executed, the SFC is in the reset state. Then the initial step (S1) becomes active, and transition T1 is evaluated. The action logic associated with step S1 is then executed.
- On the second scan, the transition logic associated with active transition T1 is evaluated and found to be true. Step S1 becomes inactive, Step S2 becomes active, and transition T2 is evaluated. The action logic associated with step S2 is then executed.
- On the third scan, the transition logic associated with active transition T2 is evaluated and found to be false. Since the transition condition is not satisfied, step S2 remains active and the action logic associated with step S2 is executed again.
- On the fourth scan, the transition logic associated with active transition T2 is again evaluated and found to be true. This time, transition T2 becomes inactive, step S2 becomes inactive. Step S3 becomes active, and transition T3 is evaluated. The action logic associated with step S3 is then executed.

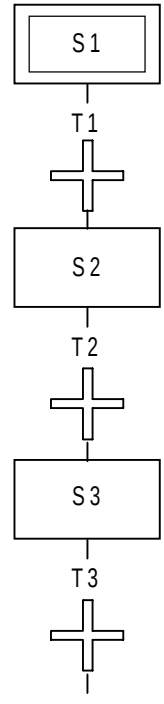


Table 2-1. Sequence of Events for the SFC Evolution Example

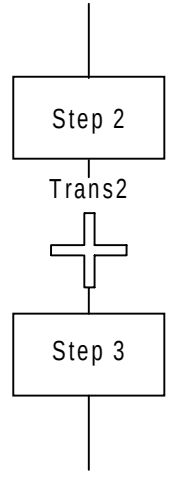
Scan	Step S1	Transition T1	Step S2	Transition T2	Step S3	Transition T3
1	Active	Evaluated	Inactive	Inactive	Inactive	Inactive
2	Inactive	Evaluated, true	Active	Evaluated	Inactive	Inactive
3	Inactive	Inactive	Active	Evaluated, false	Inactive	Inactive
4	Inactive	Inactive	Inactive	Evaluated, true	Active	Evaluated

Basic Control Structures

In the SFC topology, normal logic flow goes with gravity, that is top-to-bottom and left-to-right. The following examples illustrate the basic control structures that can be used in an SFC.

Simple Sequence

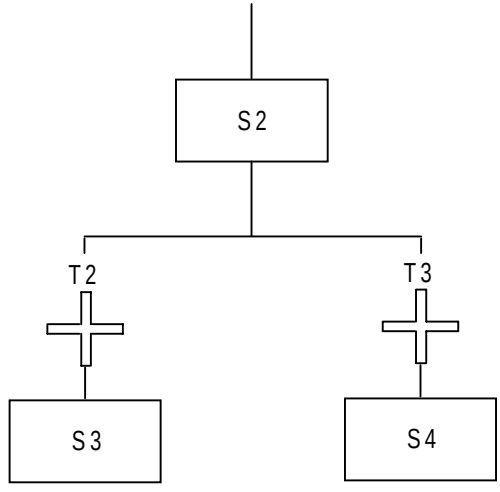
In this example, control passes from step S2 to step S3 *only if* step S2 is active and transition T2 evaluates true.



Divergence of a Selective Sequence

In this structure, there is a choice of two or more paths down which control can be passed, but only one will be selected. In the structure shown in the following illustration, control passes to step S3 *only if* step S2 is active and transition T2 evaluates true. Control passes to step S4 *only if* step S2 is active, transition T2 is not true, and transition T3 is true (left-to-right priority of transition). Exactly one branch of the selective sequence is selected. A left-to-right priority is used to determine the active branch if more than one transition evaluates true at the same time.

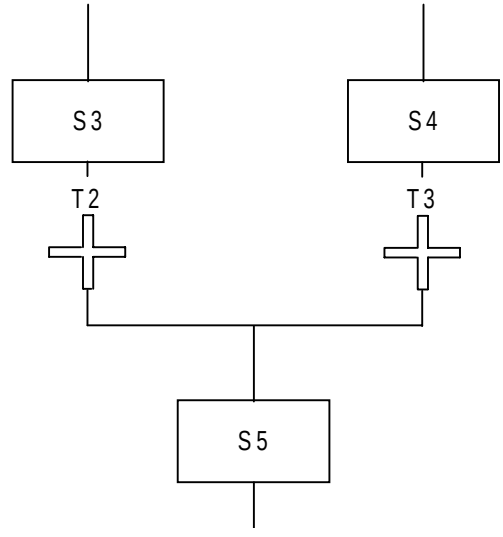
A selective divergence must be preceded by one step. The first element after a selective divergence must be a transition. The SFC Editor automatically creates a selective branch if two or more transitions are connected to a preceding step.



Convergence of a Selective Sequence

A selective convergent branch can only be preceded by transitions. It must be followed by one step. In the structure shown below, control passes from step S3 to step S5 *only if* step S3 is active and transition T2 evaluates true. Control passes from step S4 to step S5 *only if* step S4 is active and transition T3 is true.

The SFC Editor automatically creates a selective branch if one step is connected to two or more preceding transitions.

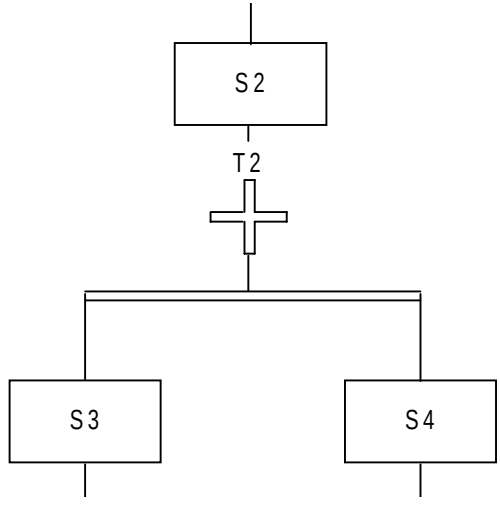


Divergence of a Simultaneous Sequence

Divergence of a simultaneous sequence provides two or more paths down which control will be passed. In a simultaneous divergent branch, more than one step will become active.

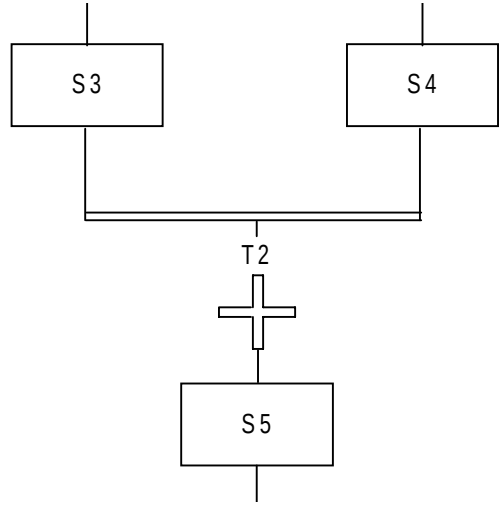
In the structure illustrated, control passes from step S2 to step S3 and step S4 *only if* step S2 is active *and* transition T2 evaluates true. Both steps will become active, although the action logic contained in one step will be executed before the action logic in the other step. The order in which steps are executed is undefined.

A simultaneous divergent branch must be preceded by one transition and followed by at least two steps. The SFC editor displays a double line to indicate a simultaneous sequence. And, it automatically creates a simultaneous branch if two or more steps are connected to a single preceding transition.



Convergence of a Simultaneous Sequence

A simultaneous convergent branch can only be preceded by step elements. It must be followed by one transition. An example of this structure is shown in the following illustration.

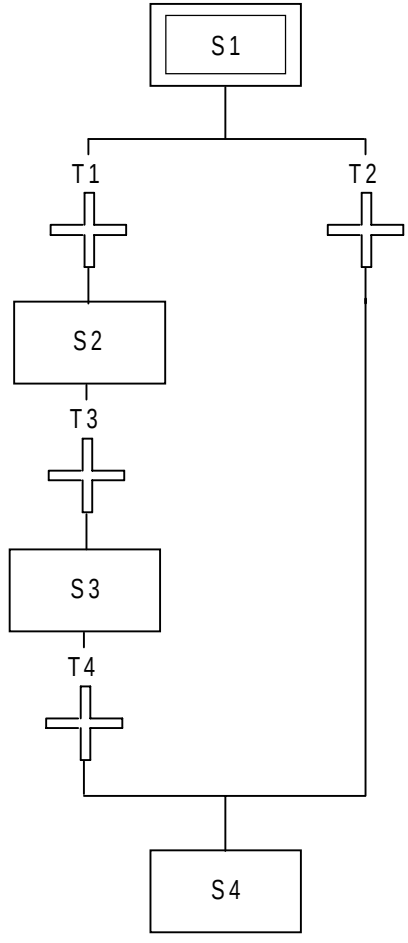


In the structure illustrated, control passes to step S5 *only if* steps S3 and S4 are both active and transition T2 evaluates true. The transition logic for T2 is only executed when all of the steps at the end of the simultaneous sequence are active. The SFC editor displays a double line to indicate a simultaneous sequence. It automatically creates a simultaneous branch if one transition is connected to two or more preceding steps.

Sequence Skip

A sequence skip structure contains a selective branch that allows a series of steps and transitions to be skipped.

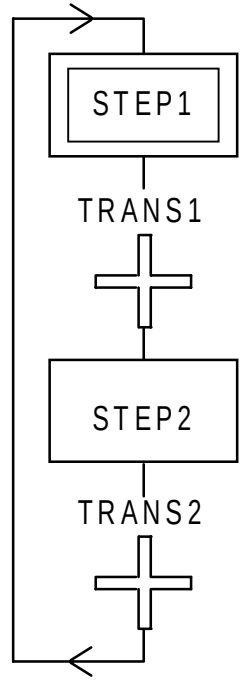
In the sequence shown below, control passes to step S2 *only if* T1 is true. If T1 is false and T2 is true, the sequence S2 through T4 is skipped and control goes immediately to S4.



Sequence Loop

A sequence loop structure uses the directed link to define a logic sequence that goes against gravity. (Normal logic flow goes from the top to the bottom of the SFC topology, or with gravity.)

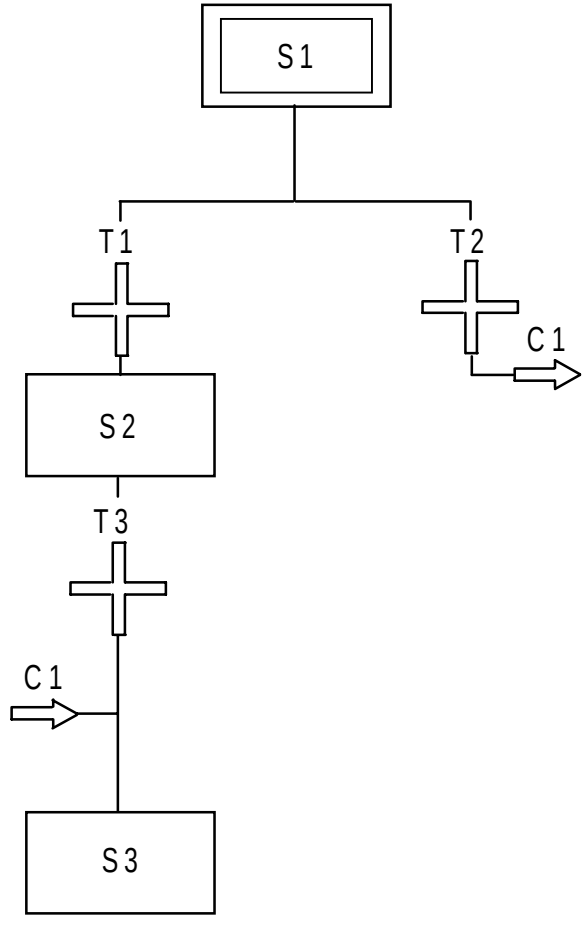
In the following figure, a directed link is used to create an iterative loop in the topology. Control passes to Step 2 *only if* Step 1 is active and Trans1 evaluates true. If Step 2 is active and Trans2 evaluates true, control returns to S1.



Jump

A jump structure uses connector elements to allow a sequence to be skipped. In the sequence shown below, control passes to step S2 *only if* T1 is active. If T1 is false and T2 is true, control goes to step S3 by way of connector C1.

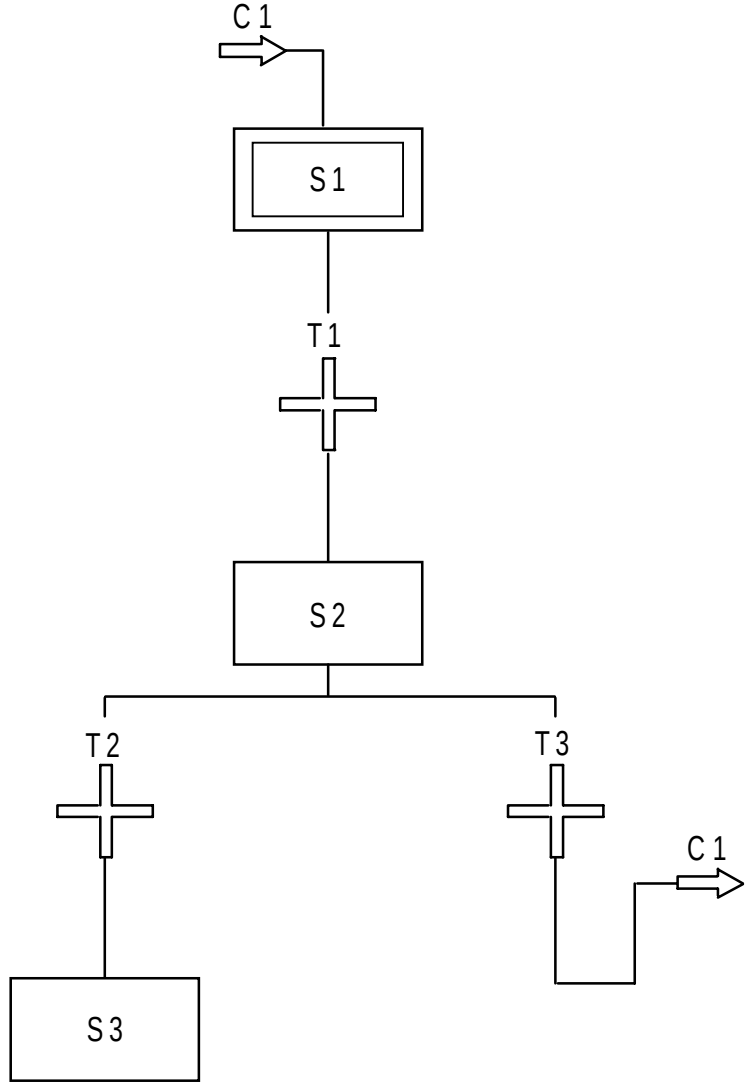
A source connector must occur immediately after a transition, as shown in the example below, where it follows T2. The destination connector must occur immediately before a step.



Cycle

A cycle structure uses connector elements to conditionally return control to the beginning of a sequence. In the sequence shown below, control passes to step S3 only if T2 is true. Control returns from step S2 to step S1 by way of connector, C1, *only if* T2 is false and T3 is true.

A source connector must occur immediately after a transition. The destination connector must occur immediately before a step.



Rules for Control Structures

An SFC should comply with the following rules:

- Only one initial step can occur in the SFC network. The initial step cannot occur within a simultaneous structure.
- Any two steps in sequence must be separated by a transition.
- Any two transitions in sequence must be separated by a step.
- Each transition must contain *only one rung* of logic which, at a minimum, sets its transition variable. For example, transition TRANS1 must contain a rung that sets the transition variable TRANS1.
- Terminal steps are not allowed in branches that converge within a simultaneous sequence . (A terminal step is one that is not followed by a transition.)
- A jump cannot transfer control into or out of any branch of a simultaneous sequence.
- Two divergent points of the same type of sequence (simultaneous or selective) can have a common convergent point. Similarly, one divergent point can have multiple convergent points. An example is shown in Figure 2-2.
- The divergence and convergence of sequences must be properly nested. For example, if a simultaneous sequence diverges after a selective sequence diverges, it must converge before the selective sequence converges. An example is shown in Figure 2-3.
- Structures must be IEC-compliant (See page 2-23).

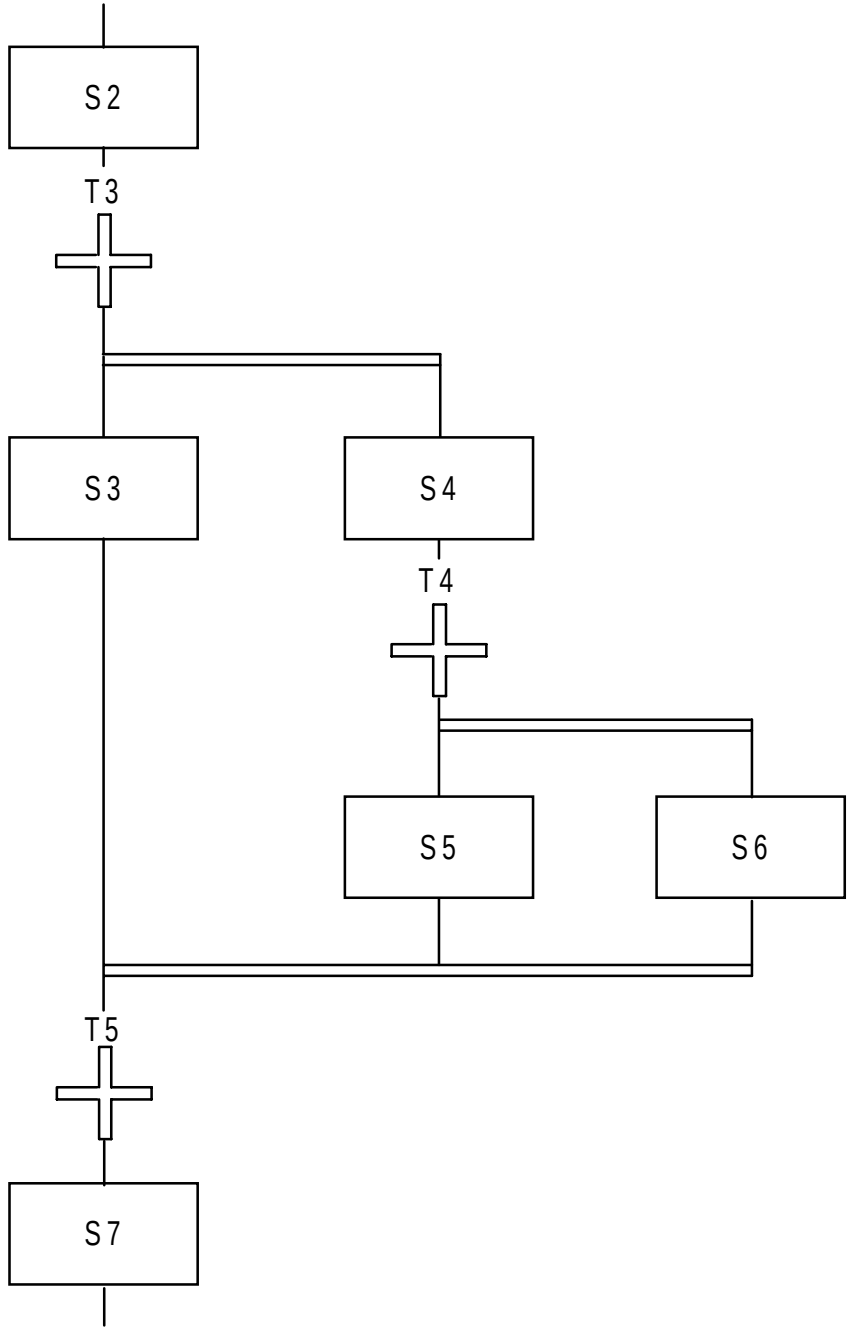


Figure 2-2. Convergence of Simultaneous Sequences

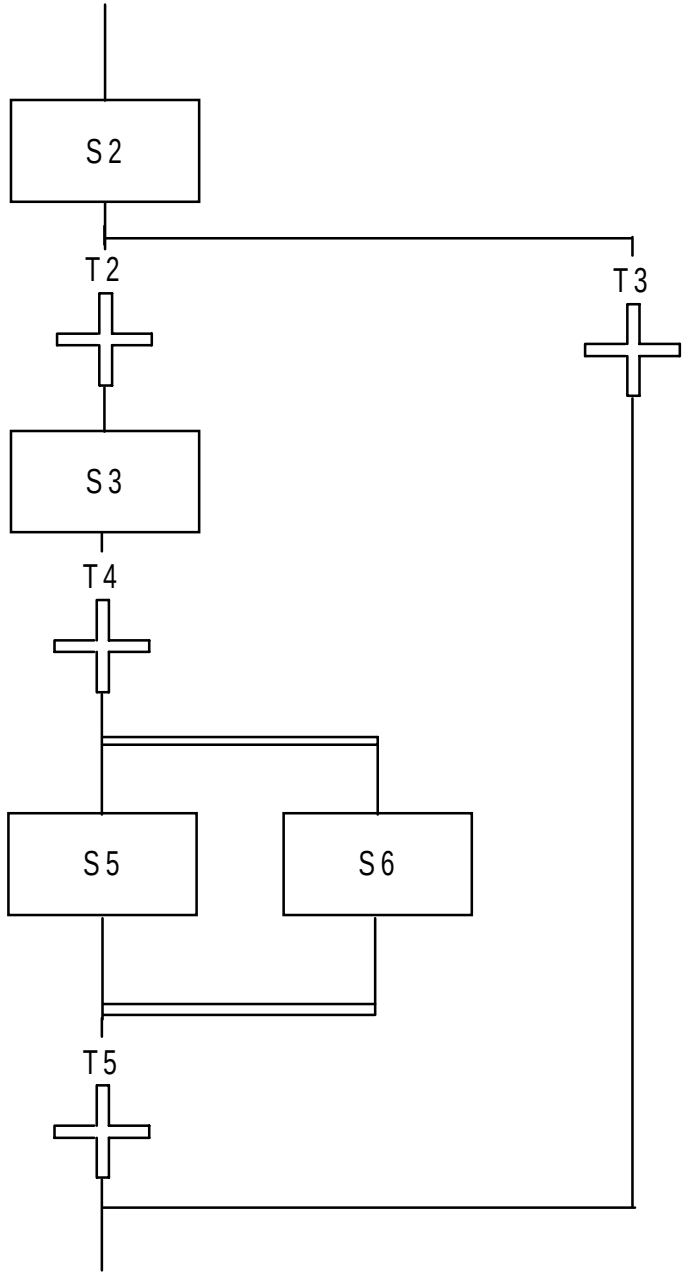


Figure 2-3. Nesting of Dissimilar Sequences

IEC Compliance

The following structures are not IEC-compliant, and cannot be created in the SFC Editor or imported. To import a Logicmaster 90 folder that contains either of these structures, first edit the structure as shown in the illustrations.

- A selective branch immediately following a simultaneous convergence (Figure 2-4).
- A simultaneous divergence immediately following a selective branch (Figure 2-5).

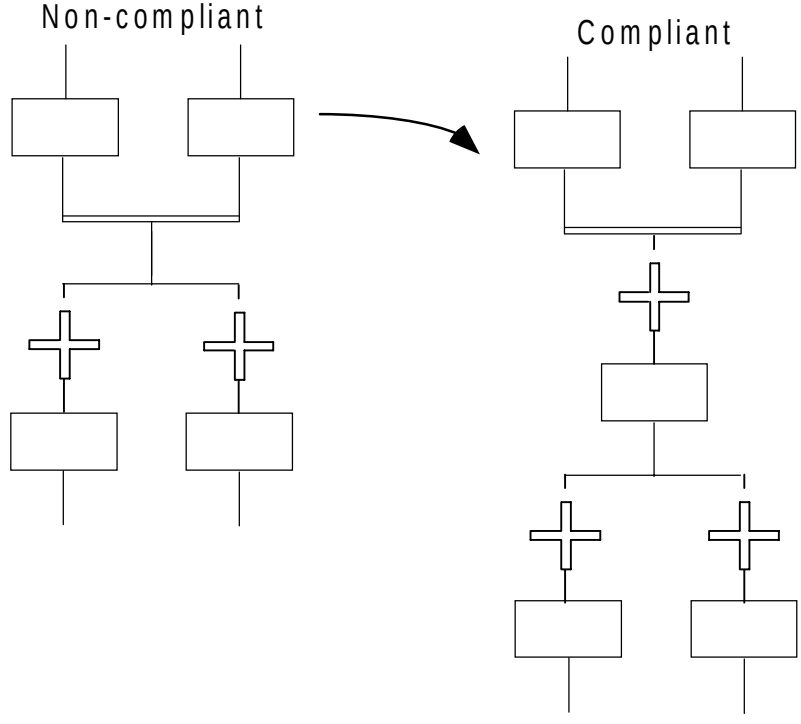


Figure 2-4. Non-IEC-Compliant Convergence

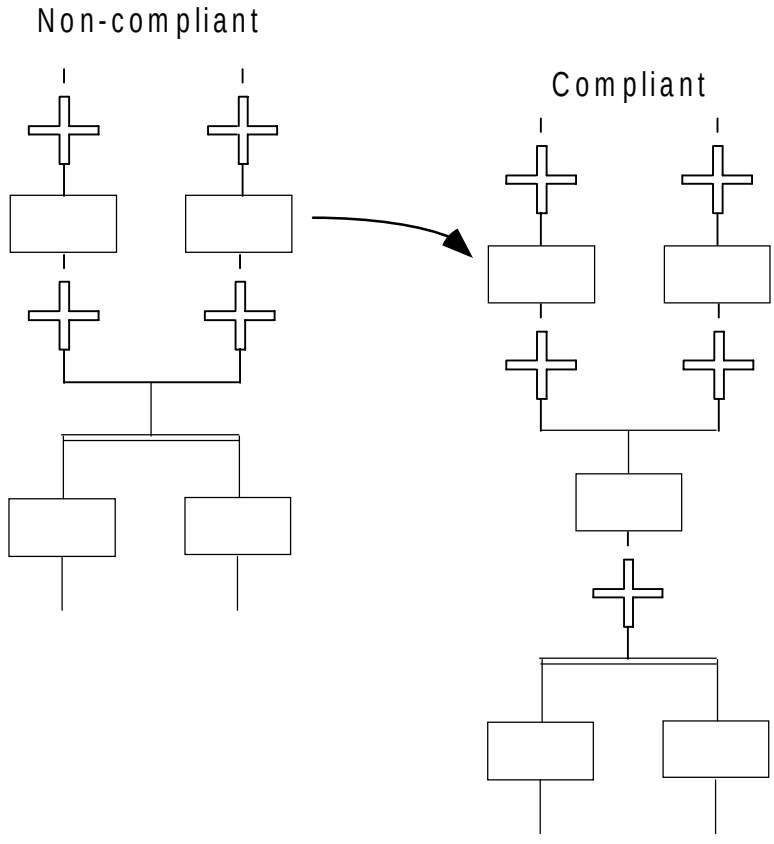


Figure 2-5. Non-IEC-Compliant Divergence

SFC Reset Function

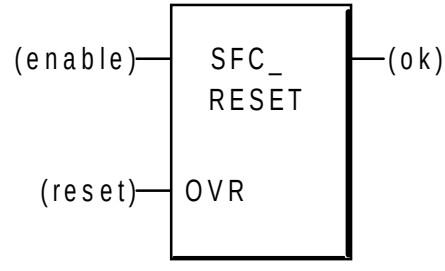
Because SFC blocks are retentive, you must use the SFC_RESET function provided in ladder diagram logic to force an SFC block to start from the initial step on power-up or after a STOP-to-RUN transition. SFC_RESET can be inserted in the action, preprocessing, or postprocessing logic. (The terminal language selected for the logic must be LD.) When the SFC_RESET function is executed, the step status values, step.x, step.t, and step.f, are cleared. You can also use the SFC_RESET function to clear override conditions on transitions (see "Forcing Transitions" in Chapter 4).

In the network evolution that immediately follows execution of SFC_RESET, all currently active steps are deactivated. This causes all the action logic associated with the active steps to be executed without power flow. Then, the initial step in the network is activated, and the activated logic is executed.

To insert an SFC_RESET instruction, go to the Terminal Language window for the selected logic. Select SFC_RESET from the ladder diagram instruction list box or toolbar, then click in the Terminal Language window to place the instruction.

Format of the SFC_RESET Function

The enable (EN) and override (OVR) inputs are used to control execution of the SFC block. The EN input is required; use of OVR is required for Series 90-30 PLCs and optional for Series 90-70 PLCs. When EN is active, the SFC_RESET function causes the SFC block to begin at its initial step and the step status values are cleared. When OVR is active, override bits for transitions are cleared.



Chapter 3 Working in the SFC Editor

The SFC Editor graphically represents the functions of a sequential automated system as a series of steps and transitions. Using the SFC Editor, you can create, modify, debug, and monitor SFC networks within the CIMPLICITY Control Equipment Workbench. An SFC block contains one SFC logic network and two optional entities, preprocessing logic and postprocessing logic. An SFC block can contain more than one SFC network.

This chapter describes how to use the SFC Editor to create an SFC network.

- Getting Started....3-2
- SFC Editor Windows....3-5
- Setting Edit and Display Features....3-14
- Building an SFC Network....3-20
- Editing an SFC Network....3-27
- Working with Preprocessing and Postprocessing Logic....3-40
- Documenting Your SFC Project....3-41
- Notes for Logicmaster 90 Users....3-43

Getting Started

To access the SFC Editor, you must specify SFC as the language for the program block (or blocks) that you create and edit within the CIMPLICITY Control Equipment Folder. An SFC network can be created in the CIMPLICITY Control _MAIN block and in sub-blocks that are created within the CIMPLICITY Control Equipment Folder. Only one language can be specified for each block; however, blocks within an Equipment Folder can have different languages.

Note

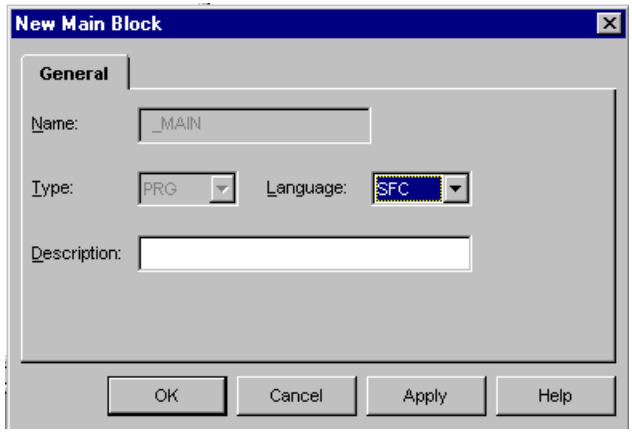
Reserved words cannot be used as file names, labels, or variable names. The following words are reserved in CIMPLICITY Control software because of their use in the SFC Editor: step*xxx*, trans*xxx*, actn*xxx*, and conn*xxx* where *xxx* is a number.

For a complete list of reserved words in CIMPLICITY Control software, refer to *Using CIMPLICITY® Control*, GFK-1295.

Creating a Main SFC Block

To specify SFC as the language for the Main program block:

1. In the Equipment Folder Browser window, expand the tree menu to display the contents of the Software Configuration.
2. With the left mouse button, double-click the Program name. (Or, click the right mouse button and choose Edit Main Block from the pop-up menu.) The New Main Block dialog box will appear.



3. Choose SFC from the Language list.

4. If you wish, you can type a description for the block in the Description field.
5. Click OK. The SFC Editor windows for the Main Block will appear.

Creating a New SFC Block

An SFC block can be created within a CIMPLICITY Control Equipment Folder as follows:

1. In the Equipment Folder Browser window, expand the tree menu to display the contents of the Software Configuration.
2. With the left mouse button, click the Program name.
3. Click the right mouse button and choose New Block from the pop-up menu.
4. In the New Block dialog box, choose SFC from the Language list. In the Type list, BLK should be selected.
5. Type a name for the new block in the Name field.
6. If you wish, you can type a description for the block in the Description field.
7. Click OK. The SFC Editor windows for the new block will appear.

Editing an Existing SFC Block

To edit an existing SFC block:

1. Go to the CIMPLICITY Control Equipment Folder Browser window, and expand the tree menu to the program block level.
2. Double click the block name. Or select the block name, click the right mouse button, and select Edit from the pop-up menu. The SFC Editor windows for the selected block will appear.

Importing a Logicmaster 90 SFC Program

SFC programs created with Logicmaster 90 can be viewed and edited in CIMPLICITY Control. When you import a Logicmaster 90 folder, you should be aware of the following:

- The positions of transitions may change. The new structure will be functionally equivalent to the original Logicmaster 90 structure.
- Connectors will be replaced with directed links. The new structure will be functionally equivalent to the original Logicmaster structure.
- Logicmaster 90 allows certain SFC structures that are not IEC-compliant. To import a Logicmaster 90 folder containing these structures, you must first edit the structures as shown in “IEC Compliance” in Chapter 2.

For more information, see “Notes for Logicmaster 90 SFC Users” on page 3-43.

SFC Editor Windows

The SFC Editor software uses three windows: Topology window, Variable Declaration Editor window, and Terminal Language window. The window that is open and selected determines which status bar, tools, and menu commands are available. (To select a window, click the mouse anywhere in it.)

The Topology window is described in detail here because it is unique to the SFC Editor. The Variable Declaration Editor and Terminal Language windows are discussed as they relate to the SFC Editor. For more detailed information on these two windows, refer to the Help for CIMPLICITY Control software and to *Using CIMPLICITY Control*, GFK-1295.

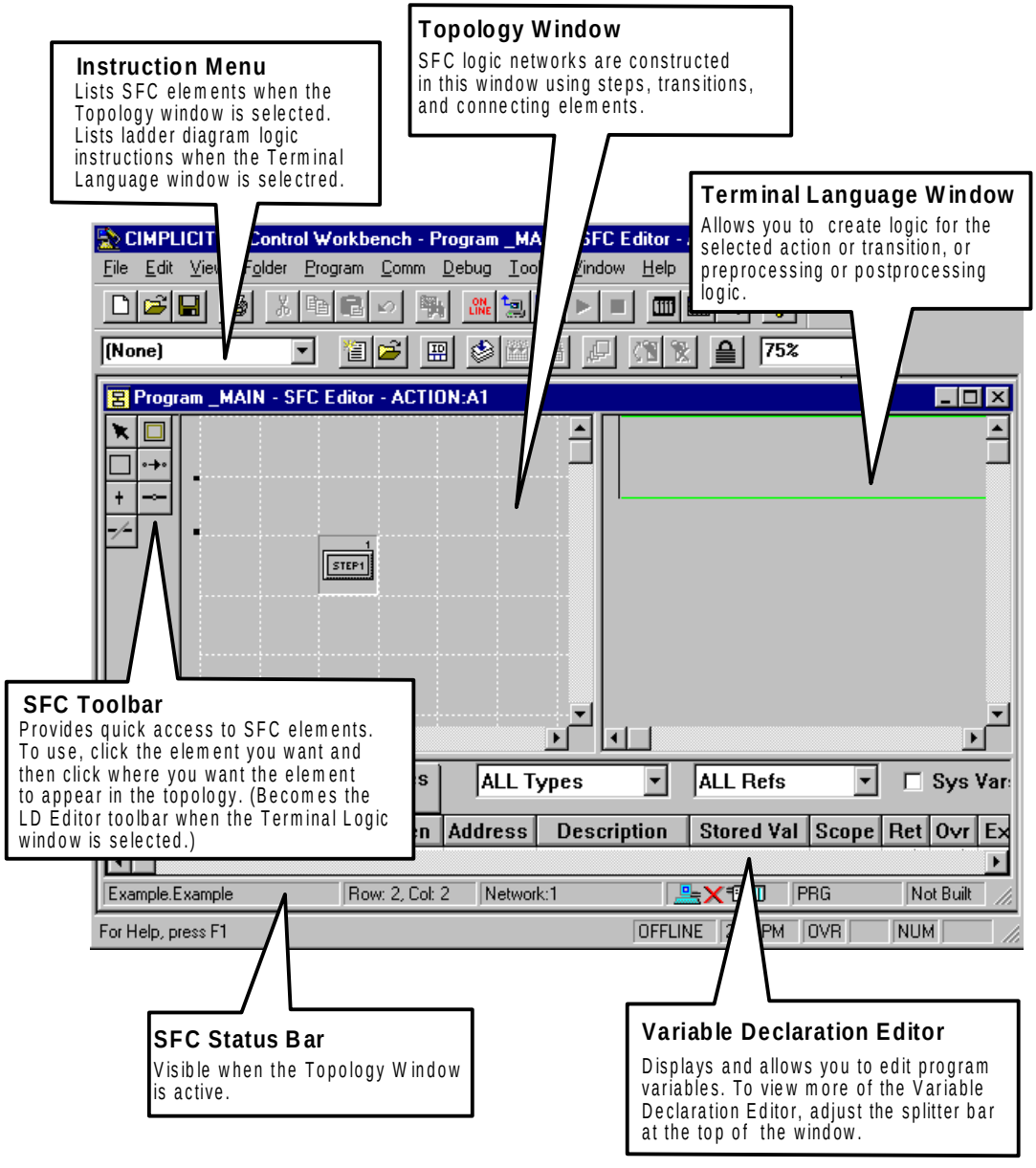

Instruction Menu
Lists SFC elements when the Topology window is selected. Lists ladder diagram logic instructions when the Terminal Language window is selectred.
Topology Window
SFC logic networks are constructed in this window using steps, transitions, and connecting elements.
Terminal Language Window
Allows you to create logic for the selected action or transition, or preprocessing or postprocessing logic.
SFC Toolbar
Provides quick access to SFC elements. To use, click the element you want and then click where you want the element to appear in the topology. (Becomes the LD Editor toolbar when the Terminal Logic window is selected.)
SFC Status Bar
Visible when the Topology Window is active.
Variable Declaration Editor
Displays and allows you to edit program variables. To view more of the Variable Declaration Editor, adjust the splitter bar at the top of the window.
Program _MAIN - SFC Editor - ACTION:A1
ALL Types
ALL Refs
Sys Var
Address
Description
Stored Val
Scope
Ret
Ovr
Example.Example
Row: 2, Col: 2
Network:1
PRG
Not Built
For Help, press F1
OFFLINE
OVR
NUM

Figure 3-1. SFC Editor Windows

Topology Window

This window displays the SFC logic network, which you construct using steps, transitions, and connectors. The SFC topology is based on a grid. Each SFC element occupies one square on the grid. The maximum grid size for a network is 255 rows x 255 columns. The Topology window is displayed with the grid on, by default. To toggle the grid display, select the View menu and choose Show Grid.

How Elements are Displayed

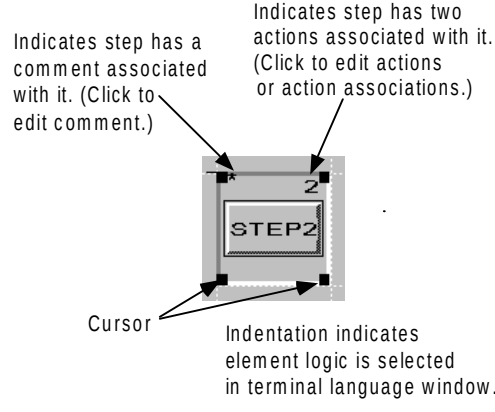
SFC Element Names

A system-generated name is automatically assigned to each element when it is created. You can assign a user-defined name to an element when it is created or at a later time, using the Properties dialog box.

The SFC Editor displays the user-defined names of the SFC elements by default. If you do not provide a user-defined name, the system-generated name will be displayed. You can choose to display system-generated names only, instead of user-defined names (see “Display Mode”).

Note

If the number of characters in the user-defined name of an SFC element name exceeds the maximum number of characters that can be displayed, a truncated name followed by a “+” is used. When you place the mouse cursor over the selected SFC element, the full user-defined name will be displayed in a tool-tip Help.


Indicates step has a comment associated with it. (Click to edit comment.)
Indicates step has two actions associated with it. (Click to edit actions or action associations.)
2
STEP2
Cursor
Indentation indicates element logic is selected in terminal language window.

Step-Action Association

A number in the upper right corner of each step indicates the number of actions associated with the step. (Action logic is programmed in the Terminal Language window of the SFC Editor.) When you left click this number, a pop-up menu that lists the associated actions appears. To edit the underlying logic, click the action name in the pop-up menu. To create a new step-action association, select New Association from the pop-up menu.

Comments

Comments can be associated with any step, transition, or action defined in the SFC topology. The SFC Editor displays the symbol * in the upper left corner of each element that has a comment associated with it. The display of symbols can be toggled on or off.

To add a comment to a selected step or transition, click the right mouse button, and choose Comment from the pop-up menu, or choose Comment from the Edit menu. To add a comment to an action, go to the Action Pool dialog box (see "Editing the Action Pool").

Logic

The logic associated with an element will be displayed automatically when the element is selected if Automatic View Logic is selected in the SFC Editor Options dialog box. To enable this feature, see "Changing SFC Editor Options." You can also display the logic associated with an element by double-clicking the element, or in the case of steps, right clicking the action number of the step and selecting an action from the pop-up list.

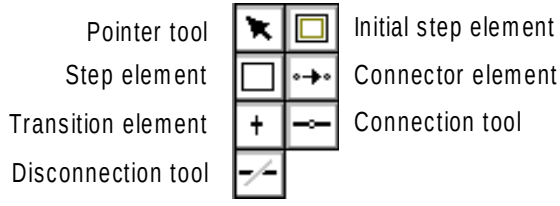
Animation of SFC Topology

Active steps and transitions are backlighted in the Topology window when the CIMPLICITY Control software is online with the PLC.

- A flashing red backlight on a step indicates that the step has an SFC time fault. (See "Setting Time Parameters" in Chapter 4.)
- A highlighted bar on a transition indicates that it is overridden.

SFC Toolbar

The toolbar contains icons that represent SFC language elements (initial step, step, transition, and connector) and editor tools (pointer, connection, and disconnection). The elements can also be selected from the drop-down list on the common Editor toolbar or the Elements dialog box. This toolbar is available only when the Topology window is selected.



SFC Status Bar

To toggle the status bar on or off, select the View menu and choose Status Bar. When the Topology window is active, the SFC status bar displays the following information:

- Context path
- Row and column location of the cursor
- Current network where the cursor is located
- Status of the connection to the target PLC

  Not Connected

  Connected

  Connected and Equal
- Program or block scope name (always BLK for SFC programs)
- Equality Status

  Not Saved — the program has not been saved

  Not Built — the program has been saved but not yet built to a binary file

  Equal — programmer is online and the program in the current folder is equal to the program in the PLC

  Not Equal — programmer is online and the program in the current folder is not equal to the program in the PLC

  Blank — programmer is online and not connected, or offline with a saved and built SFC topology

SFC Pop-up Menu

You can open the SFC pop-up menu by clicking the right mouse button anywhere in the Topology window. Menu items that do not apply to the selected element or network are grayed out. The pop-up menu contains the following items.

Table 3-1. SFC Pop-up Menu Choices

Edit	Displays the underlying logic of the selected step or transition. When you start the Edit operation on a step, logic for the step's default action is displayed.
Properties	Allows you to view and edit the properties of a step, transition, or connector. The Properties dialog box contains Language (applies to transitions and actions only), User-defined Name (optional), and Description (optional).
Comment	Opens the Comments dialog box associated with the currently selected step or transition. You can add a comment or view and edit an existing comment.
Associate	Opens the Step-Action Association dialog box, which allows you to associate actions to the selected step.
Monitor	Opens the Transition Monitor dialog box for the selected transition.
Connect Element	Activates a rubberband line that allows you to connect the selected element to other elements.
Disconnect Element	Activates a rubberband line that allows you to disconnect the selected element from other elements.
Delete	Deletes the selected SFC element. When a step is deleted, the associated actions are not deleted from the action pool unless Delete Unused Actions is selected in the SFC Editor Options dialog box.
Cut	Cuts the selected element(s).
Copy	Copies the selected element(s).
Paste	Pastes the most recently cut or copied element(s).
Paste (with Logic)	Pastes the most recently cut or copied element(s) including the underlying logic.
Insert Network Above	Inserts a new SFC network above the selected network in the current SFC block. (Available only for target PLCs that support multiple networks.)
Insert Network Below	Inserts a new SFC network below the selected network in the current SFC block. (Available only for target PLCs that support multiple networks.)
Analyze	Checks the validity of the network topology. Topology errors are displayed in the Output Log window.
Toggle	Toggles the state of the selected transition when online.
Override	Overrides the state of the selected transition when online.
Store Program Changes	When connected and online, stores only the logic program changes to the PLC. Allows you to store program changes when the PLC is running.

Key Functions

Keyboard shortcuts are available for many functions in the Topology window. These functions and the corresponding keystrokes are grouped in the following table by type.

Table 3-2. Keyboard Shortcuts

Moving Around in the Topology Window	
↑	Moves cursor up one row.
↓	Moves cursor down one row.
←	Moves cursor left one column.
→	Moves cursor right one column.
HOME	Moves cursor to the beginning of the current row.
END	Moves cursor to the end of the current row.
CTRL + ↑	Moves cursor up one network.
CTRL + ↓	Moves cursor down one network.
PAGE DOWN	Moves cursor down one window.
PAGE UP	Moves cursor up one window.
CTRL + PAGE UP	Moves cursor to the top of the window.
CTRL + PAGE DOWN	Moves cursor to the bottom of the window.
CTRL + HOME	Moves cursor to the first row of the first SFC network.
CTRL + END	Moves cursor to the last row of the last SFC network.
Editing the SFC Topology	
F2	Selects the Elements list box. (Use the up and down arrows to scroll the list, then click or press ENTER to drop the element in the SFC topology.)
CTRL + Z	Undoes the most recent delete operation.
CTRL + X	Cuts the selected element.
CTRL + C	Copies the selected element.
CTRL + V	Pastes the selected element.
CTRL + SPACE	Connects (or disconnects) the selected element. (First, turn on connection rubberbanding by pressing Alt + E and selecting Connect Element (or Disconnect Element) from the Edit menu. Then move the cursor to the element to be connected to, and press CTRL + SPACE.
DEL	Deletes the selected element.
ESC	Ends the current activity. If rubberbanding is active, turns it off.
ENTER	Edits the selected step or transition. Displays the corresponding underlying logic in the Terminal Language window.
ALT + ENTER	Opens the Properties dialog box for the selected element.

Editing Variables	
CTRL + W	Switches control between the SFC Topology and Variable Declaration Editor windows.
Editing Logic	
F6	Switches between the SFC Topology and Terminal Language windows. The logic for the selected Step or Transition will be displayed in the Terminal Language window.
F5	Switches between the SFC Topology and Terminal Language windows. Preprocessing logic will be displayed in the Terminal Language window.
SHIFT + F5	Switches between the SFC Topology and Terminal Language windows. Postprocessing logic will be displayed in the Terminal Language window.
Using File Functions	
CTRL + N	Creates a new folder. (Only one folder can be active at a time.)
CTRL + O	Opens a folder. (Only one folder can be active at a time.)
CTRL + S	Saves current program block.
CTRL + P	Prints current program block.
CTRL + F4	Closes current program block.
Using Transfer Utilities	
ALT + S	Stores your logic program to the PLC. Allows you to store program changes with the PLC running.
Using Online Functions	
F11	Overrides the state of the selected transition when online.
F12	Switches the state of the selected transition when online.
Getting Help	
F1	Accesses Online Help for the active window or dialog box.

Variable Declaration Editor Window

The Variable Declaration Editor window is located at the bottom of the SFC Editor window. If you cannot see this window, use the mouse to move the split bar so that the pane is wider.

The Variable Declaration Editor window displays the SFC variables that are defined in the logic used by the SFC network. When this window is selected, the Variable Declaration Editor functions provided by CIMPLICITY Control software can be used.

The Variable Declaration Editor window displays user-defined variable names assigned to SFC elements by default. If no user-defined name is assigned to an element, the system-generated name is displayed. System-generated names also appear in the Variable Browser dialog box so that they can be used within SFC element logic for actions and transitions.

Note

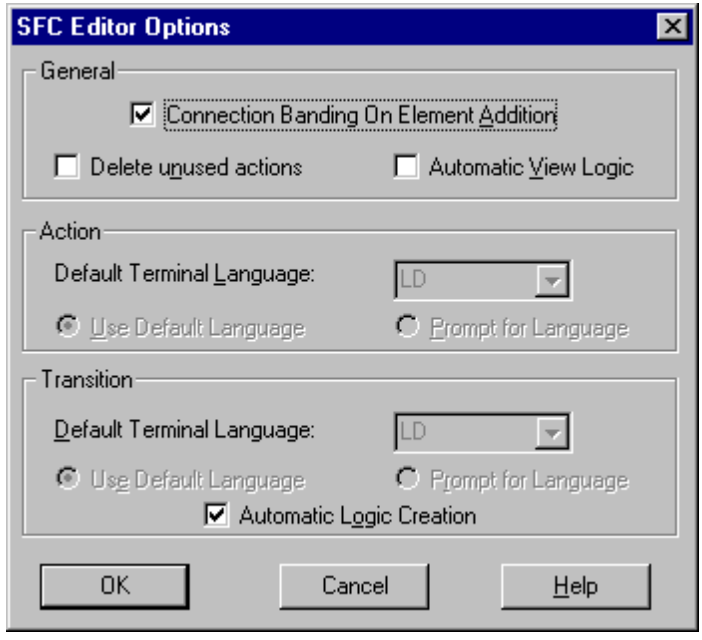
Transition and step variables are local scope, read-only variables that are handled as system variables by the Variable Declaration Editor. For convenient viewing of SFC variables, go to the Local Scope tab with the Sys Vars box checked.

Terminal Language Window

The Terminal Language window displays the underlying logic of an action, a transition condition, or the preprocessing/postprocessing logic. The underlying logic is programmed in a language that is supported by the target PLC. When the logic is programmed in Ladder Diagram language, all the functionality of the CIMPLICITY Control LD Editor can be used.

Setting Edit and Display Features

Changing SFC Editor Options



You can set options that determine how the Editor operates when you are working in the Topology window. General options determine the behavior of element type selection and connection banding. The action and transition options determine the default terminal language and whether you are prompted to change the terminal language when these elements are created.

To change SFC Editor options, select the Tools menu and choose SFC Editor. The SFC Editor Options dialog box will appear.

General

Connection Banding on Element Addition

If this box is checked, a connection band appears when the element is dropped into the topology. (Default setting is checked.)

To immediately connect the new element to another element, left mouse click an existing element, or select a new element from the toolbar and drop it into the topology. To remove the connection line, press ESC.

Delete Unused Actions

If this box is checked, actions that are associated to a step will be deleted when the step is deleted. An action that is associated to any other step will not be deleted. (Default setting is not checked.)

Automatic View Logic

If this box is checked, the editor automatically updates the Terminal Language view as the cursor is moved from one SFC element to another. Also, the Terminal Language window automatically displays the logic view for the new element when it is created. (Default setting is not checked.)

Action

If the target PLC supports multiple languages, you will be able to change the following settings. Otherwise, these settings cannot be changed.

Default Terminal Language

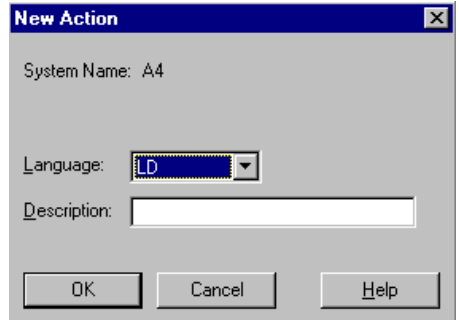
Allows you to select the default terminal language to be used during creation of actions in the empty logic. The default setting is LD (Ladder Diagram).

Use Default Language

If this option is selected, the default terminal language will be used to create actions. Either this option or Prompt for Language must be selected. Use Default Language is the default selection.

Prompt for Language

Allows you to choose a terminal language other than that specified in the Default Terminal Language field. If this option is selected, the New Action dialog box appears when a step is created because a default action is created at the same time.



Transition

If the target PLC supports multiple languages, you will be able to change the following settings. Otherwise, these settings cannot be changed.

Default Terminal Language

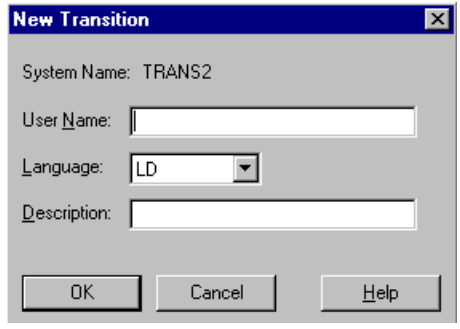
Allows you to select the default terminal language to be used during creation of transitions in the empty logic. The default setting is LD (Ladder Diagram).

Use Default Language

If this option is selected, the default terminal language will be used to create transitions. Either this option or Prompt for Language must be selected. Use Default Language is the default selection.

Prompt for Language

Allows you to choose a terminal language other than that specified in the Default Terminal Language field. If this option is selected, the New Transition dialog box will be displayed each time you create a transition.



Automatic Logic Creation

If Automatic Logic Creation is selected, the editor automatically creates default logic that sets the transition variable.

Using Lock Cursor

When Lock Cursor is enabled, the SFC element type remains selected after you drop it into the topology. This feature allows you to place the same element into the topology repeatedly without having to select it each time. The default condition for Lock Cursor is off.

To turn this feature on or off, select the Edit menu and choose Lock Cursor, or click the Lock tool on the Workbench toolbar.

Customizing the Topology Window Display

Options for customizing the Topology window include adjusting the window and grid size and selecting display modes. You can also customize the amount of information that is displayed with elements.

Grid Display Options

You can hide and display grid lines and widen grid spaces in the SFC Topology window.

Show Grid: The default view displays grid lines. To hide grid lines, select the View menu and click Show Grid. (A check mark beside Show Grid sets grid lines.)

Wide Grid: The Wide Grid view is useful when variable or reference description names are long. To set wide grid view, select the View menu and click Wide Grid. (A check mark beside Wide Grid sets this option.)

Zoom

The zoom command allows you to increase or decrease the display size in the selected window (default is 100%). To change the zoom percentage, select the View menu and choose Zoom. In the Zoom dialog box, click on one of the standard percentages or click in the Percentage field and type a number between 25 and 200. You can also adjust Zoom on the Common Editor toolbar.

Display Mode

To change the display features, with the Topology window selected, go to the View menu and choose Display Mode. The Display Mode submenu, which contains the following entries, will appear.

Name — SFC elements can be displayed with user-defined names or system-generated names. To switch between these two modes, choose Name from the Display Mode submenu.

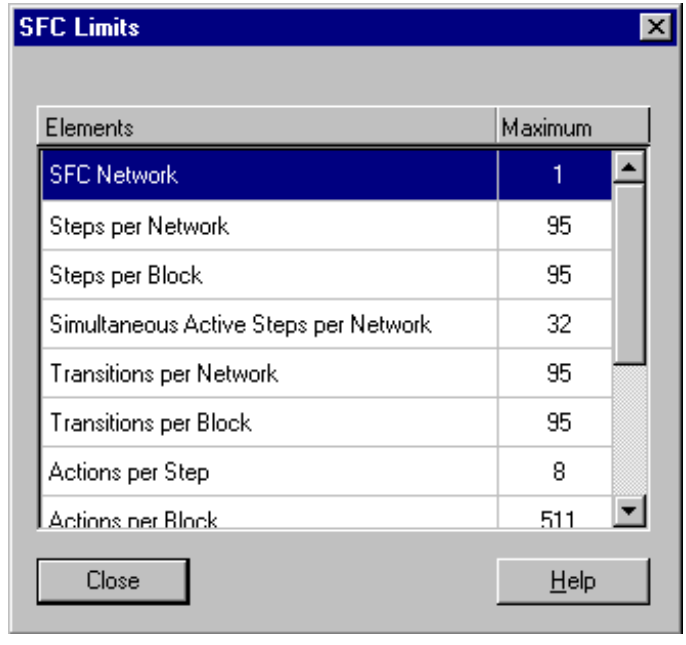
Description — The SFC elements can be displayed in the topology with their descriptions. To switch the display of the descriptions on or off, select Description from the Display Mode submenu.

Hide Comments — An asterisk is displayed next to an element that has a comment associated with it. To switch this indicator on or off, choose Hide Comments from the Display Mode submenu.

Viewing SFC Limits

The SFC Editor supports different PLC types. Each target PLC type has a unique set of constraints on the use of SFC elements: maximum number of SFC networks in a block, maximum number of SFC elements in a network, support of specific terminal languages, etc. The SFC Editor automatically handles these limitations for the current target PLC.

To view SFC limits for the target PLC, select the Tools menu and choose SFC Limits. The SFC Limits dialog box will appear.



Adding an SFC Network to the Block

The current SFC network is displayed in the Topology window. When a new SFC block is created, it contains a default network with an initial step that has one action associated. You can delete this default initial step and/or default network, but there must be an initial step in each network.

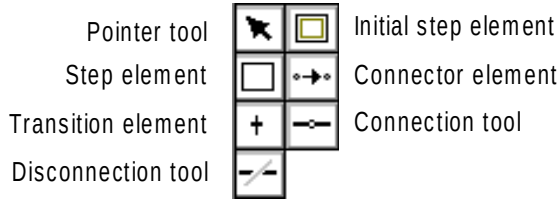
To add a new SFC network, select the Program menu and choose Insert Network Above or Insert Network Below. The new network is inserted immediately above or below the active network. In a block that contains multiple networks, all networks are evaluated and then the actions of the block are executed.

The Insert Network menu items will not be available if the current target PLC does not support multiple networks.

Building an SFC Network

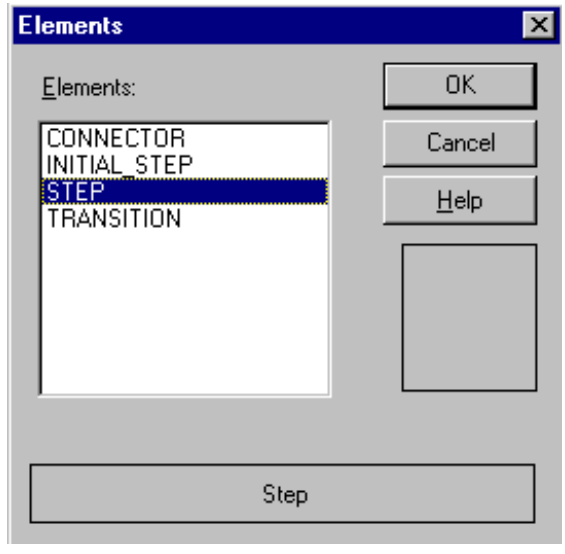
Creating Steps and Transitions

Steps and transitions are connected in the SFC topology to create the logic structure within an SFC network. You can use the SFC Toolbar or the Elements dialog box to create these elements.



To create an element using the SFC Toolbar, left mouse click on the symbol you want and then left mouse click to drop the element where you want it in the SFC topology. (Pressing ENTER will also drop the instruction at the topology cursor location.)

To access the Elements dialog box, select the Program menu and choose Elements. Select the element name from the drop-down list and then left mouse click or press ENTER to drop the element at the cursor location in the topology.



What happens when the element is dropped into the topology depends on the SFC Editor Options settings. For details, see “Changing SFC Editor Options” on page 3-14.

Note

If the Editor is in Overwrite mode, creating an element on top of a selected element will replace that element. If the Editor is in Insert mode, the new element will be created above the selected element. To switch between Overwrite (OVR) and Insert (INS) modes, press the INSERT key. (The editing mode is shown in the Workbench status bar.)

Note

You can also create elements by copying and pasting existing elements. To edit the user-defined name or description for the copied element, choose Properties from the Program menu or from the Topology window pop-up menu.

Steps

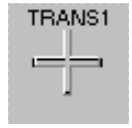
A default action containing empty logic written in the default terminal language is assigned to an initial step when it is created and to a regular step the first time you edit it. (If the Prompt for Language option is enabled in the SFC Editor Options dialog box, you will be prompted for the terminal language.) The action number is assigned by the SFC Editor in the order the actions are created. Actions can be created independently of steps (see “Creating Actions”). To be executed, actions must be associated to at least one step. Step numbers are assigned by the SFC Editor in the order the steps are created.

To edit action logic, double click on the step (edits the default action), press ENTER with the step selected, or click the action number on the upper right of the step to choose an action from the pop-up menu.

Note

Only one initial step can be defined in an SFC network. The initial step cannot appear in a simultaneous (converging or diverging) structure. The Editor allows you to construct networks that do not satisfy these conditions. However, these errors will be disallowed and logged by the Analyze function.

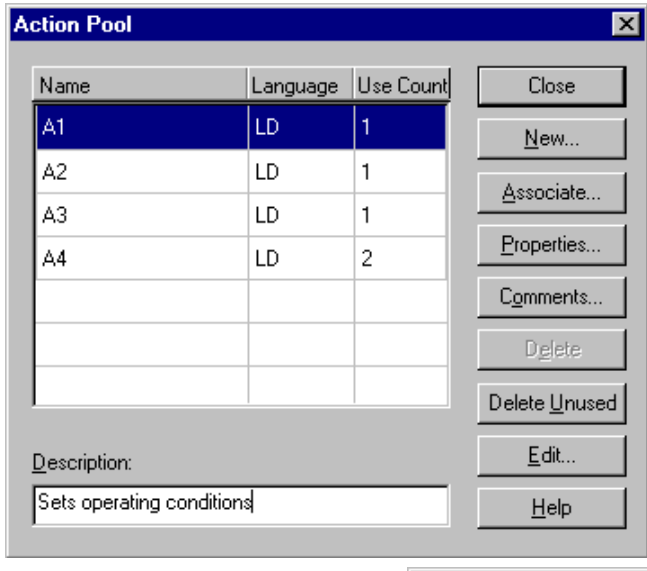
Transitions

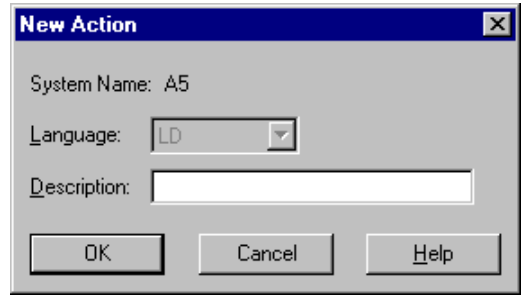


Transition numbers are assigned by the SFC Editor in the order the transitions are created. To edit logic (transition conditions) for a selected transition, choose Edit from the pop-up menu, press ENTER, or simply double click the transition.

Creating Actions

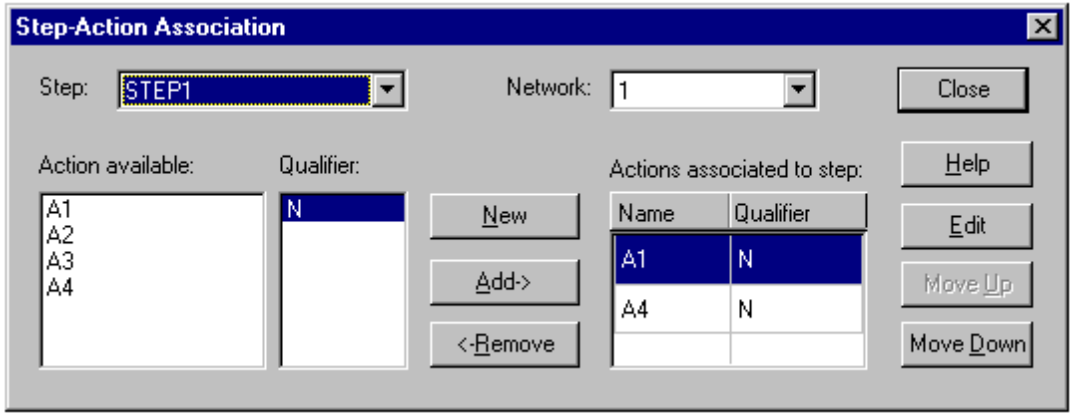
To create an action independently of a step, select the Program menu and choose Action Pool. When the Action Pool dialog box appears, click the New button. The New Action dialog box will appear. This dialog box allows you to change the default terminal language and provide a description for the new action.





To edit the logic content of the action, click the Edit button in the Action Pool dialog box. The action logic will be displayed in the Terminal Language window.

To be executed as part of the SFC, an action must be associated to a step (click Associate in the Action Pool dialog box to open the Step-Action Association dialog box). For details on editing step-action associations, see page 3-38.



Connecting SFC Elements

Usually, steps and transitions are connected using the *connection band*, creating links in which the logic flows with gravity. The connection band can also be used to create *directed links*, which flow against gravity. If you want to build a structure that includes conditional jumps or cycles, you must use a *connector* element.

Connection Band

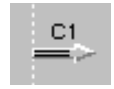
To activate the connection band, select the element you want to connect and choose the Connect tool from the toolbar, or choose Connect Element from the pop-up menu or the Edit menu. While the connection band is active, as shown at right, you can connect the new element to an existing element by left mouse clicking the existing element, or by selecting the existing element and pressing Ctrl + Space. (To remove the connection band from an element, press ESC or click the element.)

If Connection Banding On Element Addition is selected in the SFC Editor Options dialog box, a connection band appears when you place a step, transition, or connector in the SFC topology.

Disconnection Band

To activate the disconnection band, select one of the elements you want to disconnect and choose the Disconnect tool from the toolbar. You can also select the element and choose Disconnect Element from the Edit menu or the pop-up toolbar.

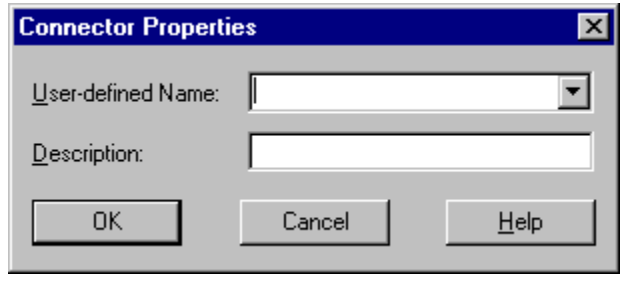
Connector



This element is used to create a jump or cycle structure. You can use the SFC Toolbar or the Elements dialog box to create a connector.

To create a connector, left mouse click the Connector icon on the SFC Toolbar, in the Elements list box, or in the Elements dialog box. (To access the Elements dialog box, select the Program menu and choose Elements.) Left mouse click or press Enter to drop the element where you want it in the SFC topology.

When the connector is dropped into the topology, the Connector Properties dialog box will appear. Typing information in this dialog box is optional. If you do not provide a User-defined Name, the SFC Editor will assign a name in sequence (C1, C2, etc.) The User-defined Name drop-down box allows you to select from existing connector names.



Directed Link

You can use a directed link to build a structure in which the logic flows against gravity. In a directed link, logic flows from a transition to a step. To create a directed link, select either the transition or step and activate the connection band. Then, position the cursor over the hot spot on the other element (the cursor will change to a directed link symbol) and click. The hot spot for directed link connection is at the bottom of a transition and at the top center of a step.

Note

If the path is not clear for a directed link that is being created, the SFC Editor will automatically place source and destination connectors instead of the directed link.

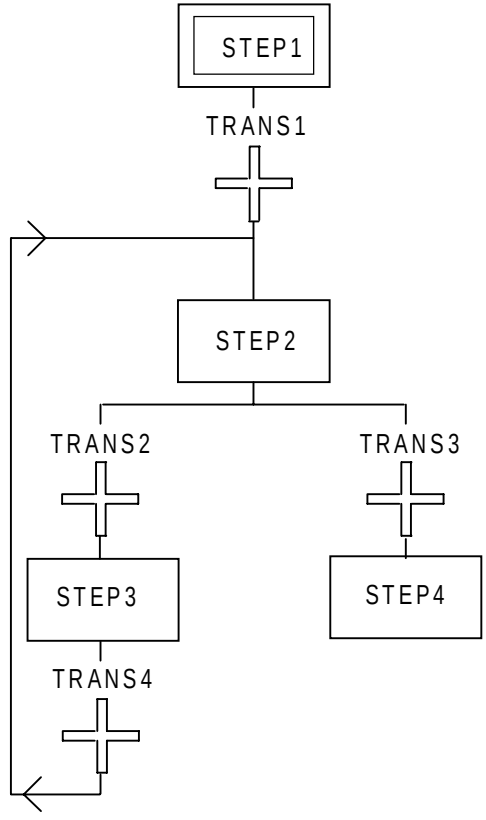


Figure 3-2. Directed Link Example

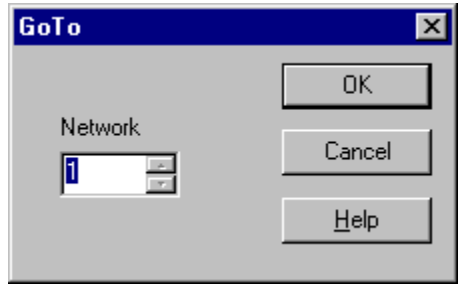
Rules for Connecting Steps and Transitions

- Connecting two steps automatically creates a transition between the steps and links the elements.
- Connecting two transitions automatically places a new step between the transitions and links the elements.
- A connector that is connected to a step is a *destination* connector. If it is connected to a transition, it is a *source* connector.

Editing an SFC Network

Navigating in the SFC Editor

To locate the part of your SFC program that you want to edit, you may need to use the Go To or Search functions. Go To applies only to programs that contain more than one SFC network.



Going to an SFC Network

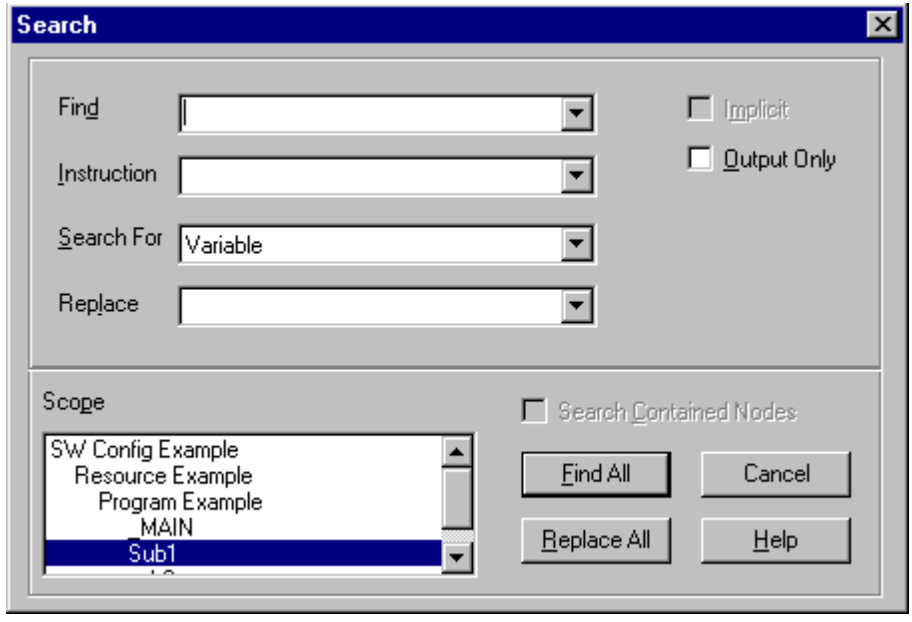
To go to a specified SFC network, select the Edit menu and click Go To. The GoTo dialog box will appear.

Select the network that you want to go to from the Network list and click OK.

Searching for SFC Elements

You can use the Search function in the SFC Editor to locate variables, reference addresses, identifiers, SFC elements, and instructions within the current equipment folder. You can also perform search and replace operations to change a variable name, reference address, identifier, or User-defined SFC element name. (You can also use the Properties dialog box, described on page 3-34, to change the User-defined Name of an SFC element.) The result of the search is displayed in the Log Output window and consists of a listing, by path, of each occurrence of the item.

To search for an item, go to the Edit menu and choose Search. The Search dialog box will appear. Edit the fields in the dialog box to specify the search parameters (field definitions are given below) and click Find All or press ENTER.



Find

Type the name of the item (specified in the Search For field) that you want to locate into this field. The default entry is the text from the last Search or Replace operation. To select from a list of recently searched items, click the drop-down arrow at the end of this field and select from the drop-down list. This field can contain:

Variable name	not case-sensitive
Reference address	example: *%T00001*
Identifier	case-sensitive
SFC Element	case-sensitive

If Instruction is selected in the Search For field, this field is grayed out.

Instruction

To limit the search by instruction, select an instruction from the drop-down list. If Instruction is selected in the Search For field, all occurrences of the instruction specified in this field will be found.

Search For

You must select one of the following items to search for:

Variable (default)	Searches for the variable name specified in the Find field
Reference Address	Searches for the reference address specified in the Find field
Identifier	Searches for the Identifier specified in the Find field

SFC Element	Searches for the SFC element specified in the Find field
Instruction	Searches for the Instruction specified in the Instruction field

Replace

If you want to perform a search and replace, select or type the replacement text here. This field is grayed out if SFC Element or Instruction is selected in the Search For field.

Implicit

This check box is available only when Reference Address is selected in the Search For field. It is grayed out for all other types of searches.

Output Only

Limits the search to outputs only. This check box is available only when Variable or Reference Address is selected in the Search For field. It is grayed out for all other types of searches.

Scope

Lists the components of the Equipment Folder to be searched. The default scope is the component from which you opened the Search dialog box. Only one component can be selected at a time.

Search Contained Nodes

This check box is grayed out for all types of searches in the SFC Editor because only one component can be selected at a time.

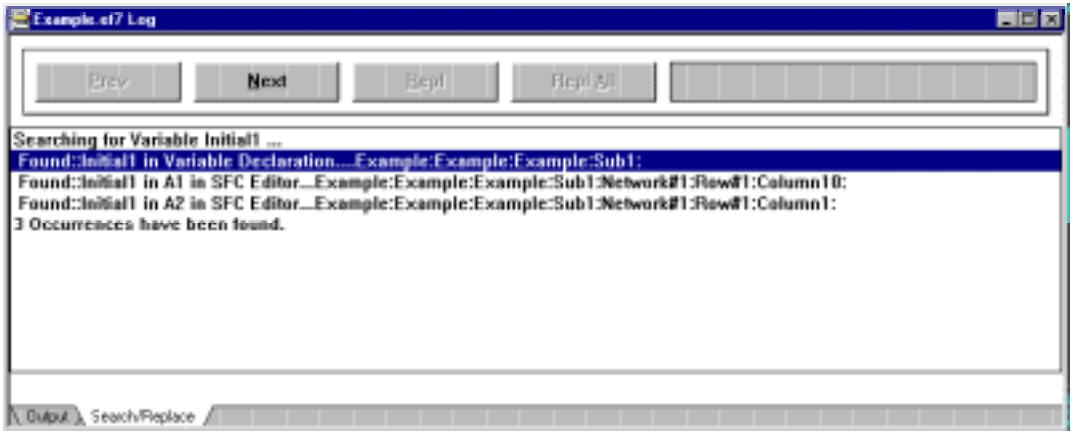
Find All

To display the results of the search in the output log and close this dialog box, click this button. At the end of the search, the Output Log window will display the total number of items found.

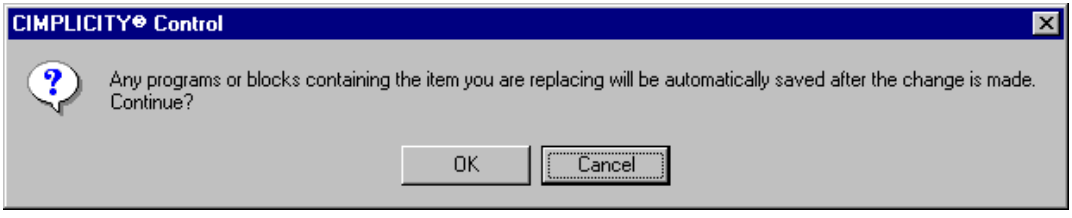
Replace All

To replace every instance of the item listed in the Find field with the item listed in the Replace field, click this button. The dialog box will close, and at the end of the search, the Output Log window will display the total number of replacements made.

The result of the search is displayed in the Log Output window includes and the path for each occurrence of the item. There will be one entry for the Variable Declaration definition and one entry for each time the item is used in the SFC network. (If you want to exclude the Variable Declaration Table and search only outputs, select the Output Only checkbox in the Search Dialog box.)



To replace a Variable name, type the text to be replaced in the Find field and type the new text in the Replace field. Click the Replace All button. The following message will appear. To continue the Replace operation, click OK or press ENTER. (You can also use change a Variable name by selecting the Variable in the Terminal Language window and opening the Properties dialog box.)


CIMPLICITY® Control
Any programs or blocks containing the item you are replacing will be automatically saved after the change is made. Continue?
OK
Cancel

Selecting Elements

To select a single element in the Topology window, left mouse click on the element. The selected element will be indicated by dots on the corners of the grid square it is on. If Automatic View Logic is selected in the SFC Editor Options dialog box, the Terminal Language window will show the logic for the selected element.

If Automatic View Logic is not selected, double click the element to edit the underlying logic. The cursor will move to the Terminal Language window and the name of the element being edited will be displayed in the SFC window title bar. (For steps with multiple actions, double clicking will open the logic of the first action associated with the step.)

To select multiple elements, drag a box around a group of elements. Using this feature, you can move, cut, copy, paste, and delete several elements at once.

Moving Elements

To use drag-and-drop editing, select an element (or a group of elements) and hold the left mouse button down until the cursor changes to . While holding the mouse button, drag the element to its new position.

Disconnecting SFC Elements

The link between a step (or initial step) and a transition can be removed by deleting the selected step or transition or by disconnecting it.

To disconnect two elements, right click one of them, then select the Disconnect tool from the SFC toolbar. You can also select Disconnect Element from the Edit menu or the pop-up menu. When the connection band appears, right click the element from which you want to disconnect.

Cutting SFC Elements

To cut a selected element, choose Cut from the Edit menu. You can also choose Cut from the pop-up menu or press CTRL+X. Cutting a step removes the element and places it on the clipboard. (If Delete Unused Actions is selected, Action(s) associated only with the step being cut will also be removed. See “Changing SFC Editor Options.”) If the element cut is a transition, its associated logic is removed.

Copying SFC Elements

To copy a selected element, select the Edit menu and choose Copy. You can also choose Copy from the pop-up menu or press CTRL+C. This operation copies the element and its associated logic to the clipboard.

- The links attached to a single step or transition are *not* copied. If multiple elements are selected, the associated links are copied.
- The Editor assigns a system-generated name to the new element when it is pasted.

Pasting SFC Elements

You can paste an element that you have cut or copied using Paste or Paste (with Logic). The two paste functions are the same for transitions and connectors. If you use Paste with a step, the associated Action(s) and logic are pasted with the step.

However, when you use Paste (with Logic) with a step, you will be prompted to choose whether a new action is created with the step or whether the existing action will be associated with the new step.

- A single element or multiple elements can be pasted. If multiple elements are pasted, the top left corner of the selection box will be pasted in the selected grid cell.
- If the Editor is in Overwrite mode, pasting on a selected element will replace that element. If the Editor is in Insert mode, the selected element and any elements to the right of it will move to the right to make room for the pasted element.
- If there is a name conflict, a new system-generated name will be assigned to the element.
- When you paste a transition, the logic is automatically changed to set the variable corresponding to the new transition.

Paste

To paste an element, first copy or cut the element, then place the cursor where you want the element to be in the Topology window, select the Edit menu, and choose Paste. You can also choose Paste from the pop-up menu or press CTRL+V. The element and its associated action and logic will be inserted.

Paste (with Logic)

To paste a step with logic, first copy or cut the step. Then, place the cursor where you want the element to be in the Topology window, select the Edit menu, and choose Paste (with Logic). You can also choose Paste (with Logic) from the pop-up menu.

When you paste the step, the Confirm Action Replace dialog box will appear (unless Delete Unused Actions is selected in the SFC Editor Options dialog box and you cut the step). If you select Create New Action (default), a new action will be created and associated with the pasted step. The new action will contain the same logic as the corresponding action in the step. If you select Over Write, the pasted step will be associated with the same action as the step from which it was created.

When you are pasting a step that is associated with multiple actions, you can select Yes or Yes to All. To be prompted for each action, click Yes. To overwrite or create new actions for all the actions associated with the step, click Yes to All.

Deleting SFC Elements

To delete a selected element or multiple elements, select the Edit menu and choose Delete. You can also choose Delete from the pop-up menu, or press the DELETE key.

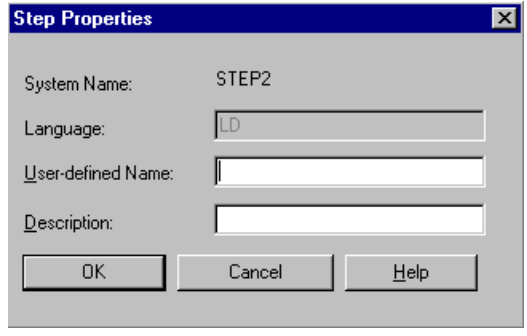
- By default, deleting a step removes it from the topology but leaves its associated actions and logic in the action pool. If Delete Unused Actions is selected in the SFC Editor Options dialog box, actions that are associated only to the step being deleted are also removed.
- If a transition is deleted, all associated logic is removed.

Undoing a Delete

To undo the most recent deletion of an element, select the Edit menu and choose Undo. Delete is the only operation that you can undo, and only if it is the most recent editing action.

Viewing and Editing SFC Element Properties

To view and edit the properties of a step, transition, connector, or action, first select the element, then select the Program menu and choose Properties. (You can also choose Properties from the pop-up menu.) The Properties dialog box for the selected element will appear.



To access properties for an action, select the Program menu and choose Action Pool. When the Action Pool dialog box appears, click the Properties button.

Note

If you add or change the User-defined Name for a step, transition, or connector, all occurrences of that element within the block will be renamed.

Editing Transition Logic

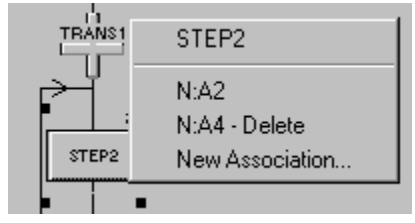
To edit the logic of a transition, double-click on the transition name in the Topology window (or, with the transition selected, press ENTER). The cursor will move to the Terminal Language window and the logic associated with the selected transition will be displayed. The transition name will be displayed on the title bar.

Note

When you are editing transition logic, the selected transition in the Topology window appears pressed in.

Editing Action Logic

You can edit the logic of an action that is associated with a step by clicking the action count number and choosing the action name from the pop-up menu.



You can also edit action logic by clicking the Edit button in the Step-Action Association or Action Pool dialog box. The cursor will move to the Terminal Language window and the logic for the selected action will appear. The step name to which the action is attached and the action name are displayed in the title bar.

Double clicking on a step that is associated with multiple actions will select the action that appears first in the Action Associated to Step list.

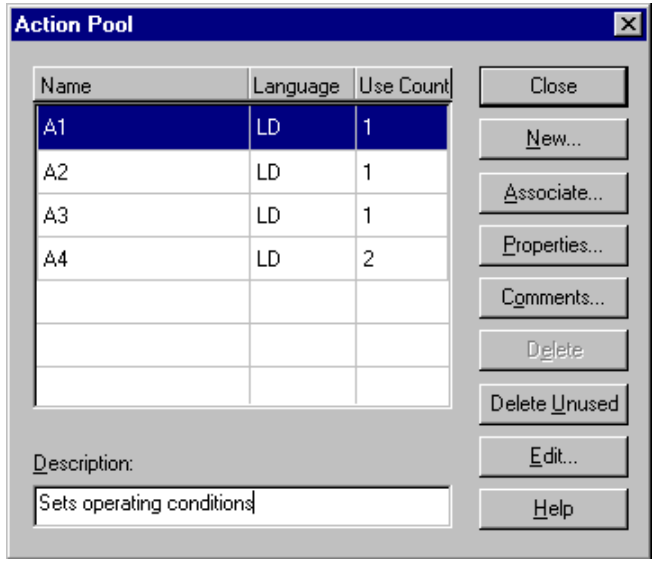
Note

When you are editing action logic, the associated step(s) appear pressed down in the Topology window.

Editing the Action Pool

The Action Pool List associated with an SFC contains a list of actions that are available for you to use in the logic. You can view this list and modify it by adding or deleting actions. You can also access the logic of a selected action from this list. (When an SFC block is created, an Action Pool that contains one empty action associated with the initial step is also created.)

To view or edit the Action Pool List, go to the Program menu and choose Action Pool. The Action Pool dialog box will appear.



The Action Pool dialog box contains the following information:

Name

Indicates the system-generated name of the action.

Language

The language in which the action is programmed.

Use Count

Indicates the number of times the action is used within the SFC block.

Close

Saves the changes that you have made to the Action Pool list and closes the dialog box.

New

Displays the New Action dialog box. The number of actions that can be added to an SFC network is limited by the maximum number of actions supported by the target PLC.

Associate

Displays the Step-Action Association dialog box, which allows you to associate the action to a step.

Properties

Displays the Properties dialog box for the selected action.

Comments

Allows you to edit or add a comment for the selected action.

Delete

Deletes the selected action if it is not being used in the block (not associated with a step). A confirmation dialog will be displayed.

Delete Unused

Deletes actions that are not associated with a step. A confirmation dialog box will be displayed.

Edit

Closes the Action Pool dialog box and places the cursor in the Terminal Language window. Logic associated with the selected action will be displayed.

Description

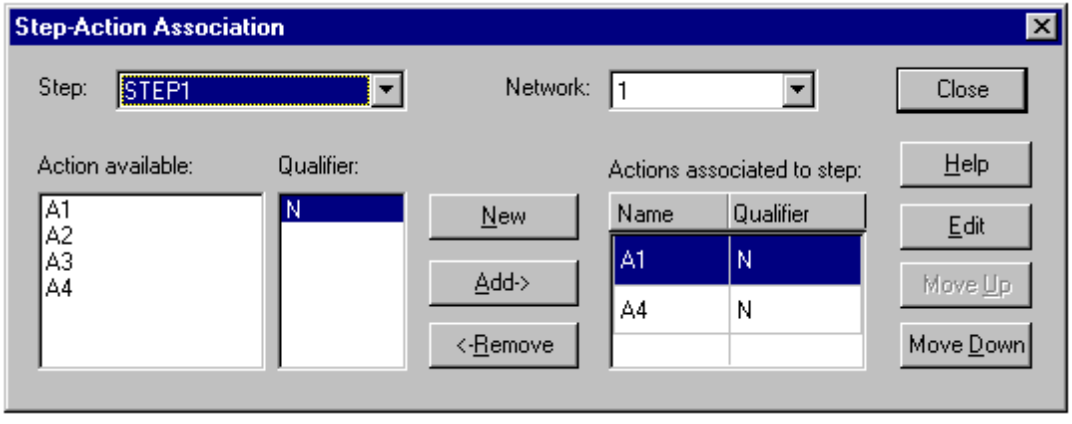
You can type or edit a description for the selected action in this box.

Editing Step-Action Associations

You can use the Step-Action Association dialog box to create new actions, associate one or more actions to a step, and remove actions from a step.

You can open the Step-Action Association dialog box in the following ways:

- Select the Edit menu and choose Associate
- Right click on a step and choose Associate from the pop-up menu.
- Click the action count number next to a step and choose New Association from the pop-up menu.
- Choose Associate from the Action Pool dialog box.



The Step-Action Association dialog box contains the following information:

Step

Indicates the selected SFC step. This list box can be scrolled to view all steps in the currently selected network. If a step is selected when you open the Step-Action Association dialog box, it appears in the Step box. (If no step is selected, the initial step is displayed.) The drop-down list presents steps in alphabetical order. User-assigned names are listed for steps that have them.

Network

Lists the networks in the SFC block. Use the drop-down list to select a network.

Action available

Lists the system-generated names of actions that are defined for the current SFC block.

Qualifier

The default action qualifier, N (non-stored — null), is specified for each step-action association.

Actions Associated to Step

Name

Lists the actions associated to the current step.

Qualifier

Lists the qualifier specified for each action that is associated to the current step.

New

To create an action, click this button. The New Action dialog box will appear. When you click the OK button in the New Action dialog box, the new action will be added to the Action available list.

Add

To assign a new action to the current step, first select the action and a qualifier, then click the Add button. Actions are added to the end of the list in the Actions Associated to Step list. The number of step-action associations is limited by the maximum number allowed by the PLC target.

Remove

To delete a step-action association, select it in the Actions Associated to Step list and click Remove. Neither the action nor the step will be removed from the block.

Edit

To edit the logic of the selected action, click the Edit button. The logic will be displayed in the Terminal Language window and the Step-Action dialog box will be closed.

Move Up

To move the selected action up in the Actions Associated to Step list, click this button.

Move Down

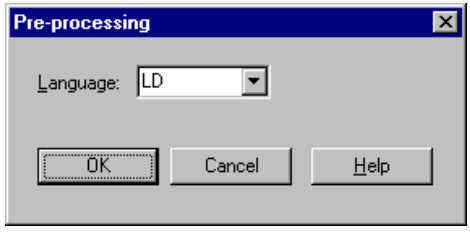
To move the selected action down in the Actions Associated to Step list, click this button.

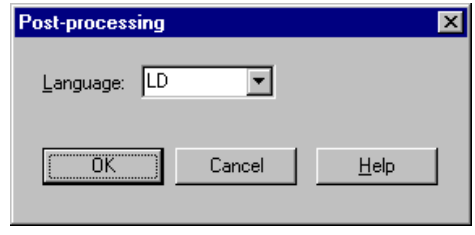
Working with Preprocessing and Postprocessing Logic

Preprocessing and postprocessing logic provides a means for programming operations that must be completed outside (before or after) the SFC. For a description of the PLC scan cycle, which includes preprocessing, sequential processing, and postprocessing, refer to Chapter 2.

Creating Preprocessing/Postprocessing Logic

To create preprocessing logic or postprocessing logic, select the Program menu and choose Pre/Post. Choose Preprocessing Logic or Postprocessing Logic from the Pre/Post submenu. (Or, with the Topology window selected, press F5 to choose preprocessing logic; press SHIFT + F5 to choose postprocessing. The Preprocessing or Postprocessing dialog box will appear. This dialog box allows you to change the default terminal language for the pre- or postprocessing logic. When you click OK, the cursor will move to the Terminal Language window.



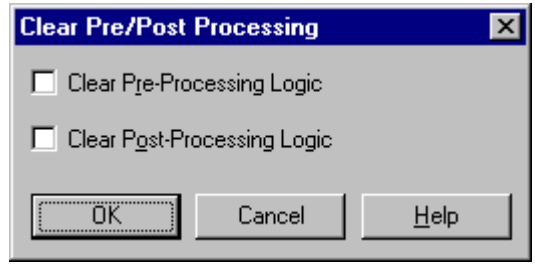


Editing Preprocessing/Postprocessing Logic

To edit preprocessing logic (or postprocessing logic), select the Program menu, choose Pre/Post, and then choose Preprocessing Logic (Postprocessing Logic). The cursor will move to the Terminal Language window and the selected logic will be displayed.

Clearing Preprocessing/Postprocessing Logic

To clear preprocessing logic or postprocessing logic, go to the Program menu, select Pre/Post, and then choose Clear. When the Clear Pre/Post Processing dialog box appears, you can choose to clear preprocessing logic, postprocessing logic, or both. (This menu item will not be available if neither preprocessing nor postprocessing logic has been created.)



Documenting Your SFC Project

By documenting your program through descriptions, comments, and summary information, you can make it easier to modify and troubleshoot.

You can provide documentation at several levels. Block summary information applies to the entire SFC block. Descriptions can be used to annotate steps, transitions, connectors, and actions within an SFC network. Comments can be used to provide additional details about steps, transitions, and actions in an SFC network.

You can use the documentation features provided by the terminal language to describe the underlying logic for actions and transitions. For example, LD allows you to add a comment to each rung in the ladder logic. Although comments are not available for preprocessing and postprocessing logic as a whole, you can document this logic by providing comments at the terminal language level.

Comments

To edit the comment associated with a step or transition (or to add a comment), select the element, then select the Program menu and choose Comment. You can also select the element and choose Comment from the pop-up menu. The Insert Comment window will be displayed.

To edit the comment for an action, with the Topology window selected, go to the Program menu and choose Action Pool. When the Action Pool dialog box appears, select the action that you want to annotate and click the Comments button.

Description

You can provide or edit the description for any selected element or action in its Properties dialog box. For a selected step or transition, select the Program menu and choose Properties. You can also select Properties from the pop-up menu, or press ALT + ENTER. To edit the description for an action, with the Topology window selected, go to the Program menu and choose Action Pool. When the Action Pool dialog box appears, select the action that you want to describe and click the Properties button.

Summary

Summary information can be provided to describe an entire SFC block. To access the summary for the block, select Header from the Program menu, then click the Summary button in the Header dialog box. Or, go to the Workbench window, select the Edit menu, and choose Summary from the Program Options submenu.

Storing Your SFC Program to the PLC

You must first establish communication and go online with the PLC before performing a Store, Load, or Verify. To access these operations, select the Tools menu and choose Transfer Utilities. In most cases, you must put the PLC in Stop mode before storing to the PLC. For details on storing Equipment Folders, refer to "Transferring Equipment Folders" in *Using CIMPLICITY Control*, GFK-1295.

The following changes can be stored in Run mode:

- Row and column changes for elements. (No other topology changes can be stored in Run mode.)
- Changes in terminal language logic.
- Step minimum and maximum time changes. (Changes to step time base cannot be stored in Run mode.)

For Series 90-30 PLCs, some changes are stored automatically in Run mode. (These are referred to as "word-for-word" changes.) These changes are not saved automatically to the programmer; you must save them manually.

A word-for-word change consists of any change that does not alter the length of the block code, requiring a Block Edit. Examples of this type of change are:

- Changes to a coil address
- Changes to a reference address

Notes for Logicmaster 90 Users

CIMPLICITY Control provides a Graphical User Interface for programming the Series 90 PLCs that is very different from Logicmaster 90. However, the basic elements for creating SFC networks are the same in the two packages. The following paragraphs highlight major differences between the SFC Editors in the two packages.

IEC-Compliant Structures

Logicmaster 90 allows certain SFC structures that are not IEC-compliant. These structures cannot be created in CIMPLICITY Control, but can be imported in a Logicmaster 90 folder or copied and pasted from a library. For more information, see "IEC Compliance" in Chapter 2.

Automatic Creation of Branches

The SFC Editor automatically creates a simultaneous or a selective branch, depending on whether the branch is created following a step or a transition. In Logicmaster 90, you must insert these branches as separate elements.

Grid Organization

In CIMPLICITY Control, the maximum grid size for an SFC network is 255 rows x 255 columns. In Logicmaster 90, an SFC network is limited to 128 rows and 8 columns. Also, in Logicmaster 90, transitions can be placed only in odd-numbered rows and steps only in even-numbered rows. In CIMPLICITY Control, there are no such restrictions as to the placement of elements.

Drag-and-Drop Editing

This feature allows you to easily move elements in the SFC topology. To use drag-and-drop editing, left mouse click the element and hold the mouse button down until the cursor changes to . While holding the mouse button, drag the element to its new position.

Selection of Multiple Elements

You can select a group of elements by dragging a box around them. In Logicmaster 90, you can select only one element at a time.

Directed Links

CIMPLICITY Control provides directed links for creating SFC structures that flow against gravity. (These structures can also be created using connectors.)

Step Timer Updates

Logicmaster 90 supports word-for-word updates to step timer values. In CIMPLICITY Control, you must use ALT+S to store new step timer values to the PLC.

Compatibility

SFC programs that are created using CIMPLICITY Control cannot be imported into the Logicmaster 90 package. However, SFC programs created in Logicmaster 90 can be imported by CIMPLICITY Control.

Chapter 4 Troubleshooting

Several features are available for use in debugging SFC networks. The Analyze function checks the network topology. Global checks are provided by the terminal language used to program the underlying logic of actions, transitions, preprocessing logic, and postprocessing logic. You can set step time and time base parameters for diagnostic use when the PLC is online. You can also include the SFC_RESET function in the Ladder Diagram logic to activate the initial step and deactivate all other steps and transitions in the network (see “SFC_RESET Function” in Chapter 2).

When the PLC status is online, you can monitor transition conditions and toggle or override transition values. You can set step time parameters for monitoring the duration that steps are active. And, evolution history can be recorded for steps in your network.

- Analyzing SFC Networks 4-2
- Setting Time Parameters 4-2
- Monitoring Transition Conditions 4-6
- Forcing Transitions 4-9
- Recording and Viewing Evolution History.....4-10

Analyzing SFC Networks

Topology errors are detected by the SFC Editor during the analysis of the SFC networks. To perform these checks during the edit session, select the Program menu and choose Analyze. The Analyze function is performed automatically when you save your program.

Errors generated by the Analyze function are displayed in the Output Log window. If no errors are generated, the message “Analysis complete. 0 errors found” is displayed in the status bar.

Setting Time Parameters

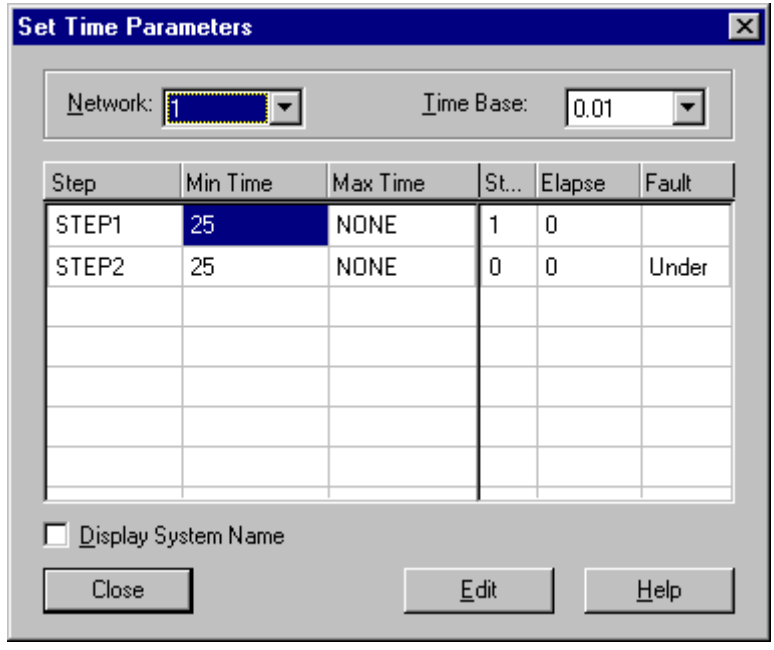
You can set the time base for the current SFC block, and minimum and maximum times for each step in the current block. When the programmer status is Online and Equal and your program is running, the Set Time Parameters dialog box records and displays elapsed time and time faults for active steps.

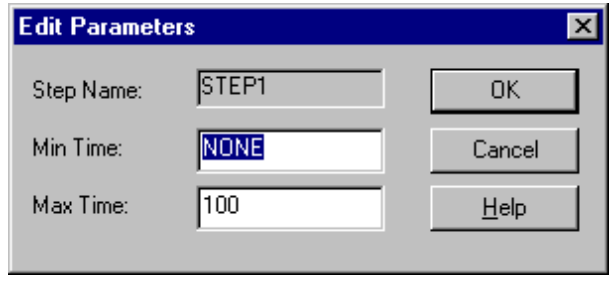
To view time or edit time parameters for the current SFC block, select the Debug menu and choose Set Time Parameters. The Set Time Parameters dialog box will appear. You can edit a selected step parameter by typing directly in its field, by scrolling the values, or by clicking the Edit button to display the Edit Parameters dialog box.

When the programmer status is Online and Equal, and your program is running, this window displays the elapsed time (the value of *stepname*.t) for steps that have been active and indicates whether the step is *faulted* (that is, the maximum step time was exceeded or the minimum step time was not attained).

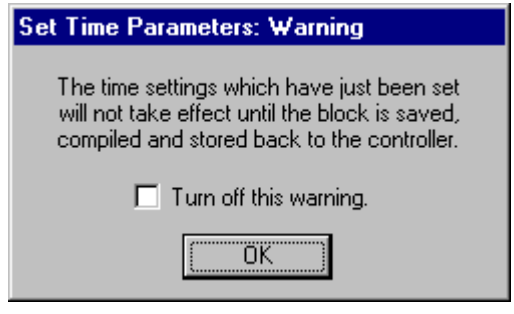
Note

If the PLC status is equal to the programmer, step times can be changed online (Run mode stores supported). Changes to the time base must be done offline (Stop mode store only).





If you edit Min or Max Time when the status is Online and Equal, and Running, the animation for the step will be stopped and the Set Time Parameters: Warning dialog box will appear. After you close the Set Time Parameters dialog box, press Alt + S to store the changes to the PLC.


Set Time Parameters: Warning
The time settings which have just been set will not take effect until the block is saved, compiled and stored back to the controller.
Turn off this warning.
OK

The Set Time Parameters dialog box contains the following information:

Network

Allows you to select the network within the current SFC block.

Time Base

Allows you to select the time base for the selected SFC network. Time base values can be 0.01, 0.1, or 1 second, depending on the capabilities of the target PLC. To change the time base, use the scroll bar, the ↑ and ↓ arrow keys, or type the value.

Table

Step

Displays the current step name. (To display system-generated names only, click the Display System Name box).

Min Time

Displays minimum step time in time base units. To edit this parameter using the scroll list, click on the cell in the table. To type in a new value, click the Edit button. (Typing 0 changes the value to NONE.)

Max Time

Displays maximum step time in time base units. To edit this parameter using the scroll list, click on the cell in the table. To type in a new value, click the Edit button. (Typing 0 changes the value to NONE.)

State

When the programmer is online, displays the state of the step.

Elapse Time

When the programmer is online, displays the elapsed time (value of *stepname*.t) of the current step.

Fault

When the programmer is online, indicates whether the elapsed time of the current step fell outside the time parameters (value of *stepname*.f is set to 1). OVER indicates the step exceeded the maximum time. UNDER indicates the step did not attain the minimum time.

Display System Name

When this box is checked, steps will be listed by system-generated names only.

Edit

To edit minimum and maximum time parameters for the selected step, click this button. The Edit Parameters dialog box will appear.

The Edit Parameters dialog box allows you to edit time parameters for the selected step. This dialog box contains the following information.

Step Name

Displays the current step name.

Minimum Time

Allows you to edit the minimum step time.

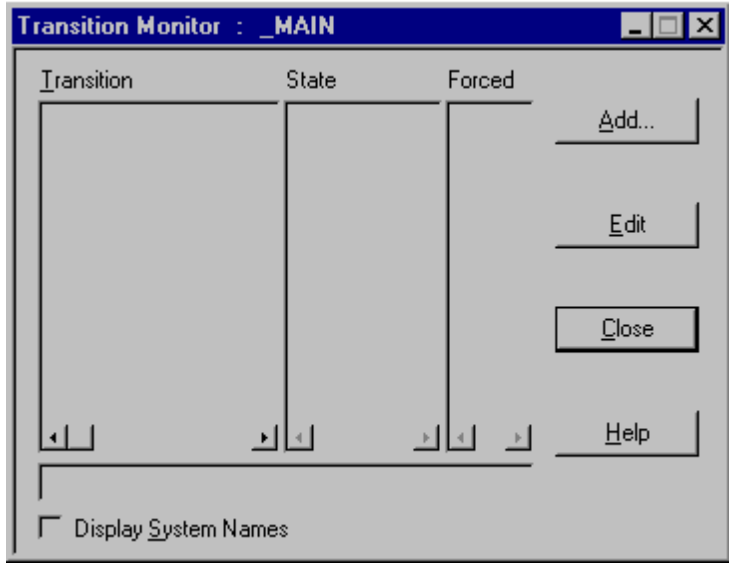
Maximum Time

Allows you to edit the maximum step time.

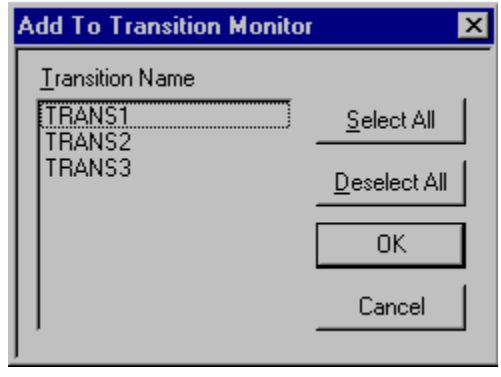
Monitoring Transition Conditions

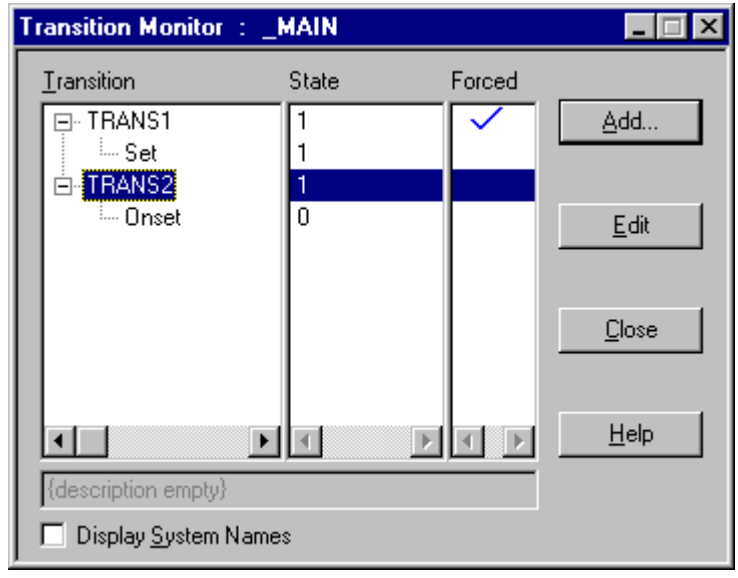
When the programmer is online and equal, you can use the Transition Monitor window to view and evaluate transitions and their corresponding variables.

To open the Transition Monitor window, select the Debug menu and choose Transition Monitor. An empty Transition Monitor window will appear. Or, select the transition in the topology, right click, and select Monitor from the pop-up menu.



To add transitions to the Transition list, click the Add button. The Add to Transition Monitor dialog box will appear. Select the transitions that you want to monitor and click OK. You can select individual transitions by clicking on them. You can also select transitions by pressing Tab to select the Transition Name list, using the up and down arrow keys to choose a transition, and then pressing the Space key to highlight your selection.





The Transition Monitor window is updated with each evolution of the SFC network and shows the states and forced status of each transition being monitored.

The Transition Monitor window contains the following information:

Transition

Lists the names of the transitions currently being monitored. The list is always empty when this window is initially opened. If the Display System Names box is checked, only system-generated names are displayed. To see a list of variables used in a transition, including their states and forced status, click the + shown to the left of the transition name.

State

Displays the current state (or value) of the transition. Updated with each evolution of the SFC network.

Forced

Indicates whether the transition is forced (see “Forcing Transitions”).

Display System Names

To display only system-generated names for transitions, check this box.

Add

To add transitions to the list, click this button. The Add to Transition Monitor dialog box will appear. Select the transitions you want to monitor from this dialog

box and click OK. (To clear the Transition list, close the Transition Monitor window and reopen it.)

Edit

To edit transition logic, select a transition and click this button. The logic for the selected transition will be displayed in the Terminal Language window.

Close

Closes the dialog box, clears the list of transitions, and stops the monitoring process.

Description

If you have provided a description in the Transition Properties dialog box, it will be displayed in the box below the table.

Forcing Transitions

If the programmer is online and equal to the PLC, the states of the transitions in the SFC can be changed, regardless of the transition logic. This can be done by toggling or by overriding the transition state.

Toggle Transition

To toggle the state of a transition when the PLC is online, select the transition, go to the Debug menu and choose Toggle (or press the F12 key). The effect of toggling a transition may last only one sweep. On the next sweep, the transition will be evaluated and the toggled value could be replaced if it is not manually overridden. An enabled (active) transition is shown with backlighting in the Topology window. To keep the transition in a toggled state, use the Override Transition function first.

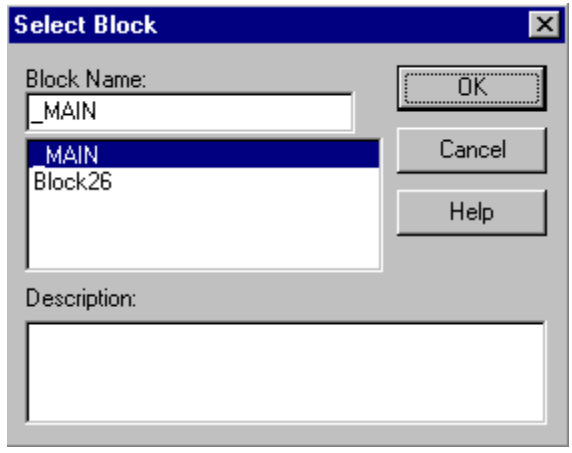
Override Transition

To force the transition to remain in a toggled state, you must first override the transition. (The PLC must be online to do this.) To override the transition, select the Debug menu and choose Override (or press the F11 key). An overridden transition is shown with a bar through it in the Topology window.

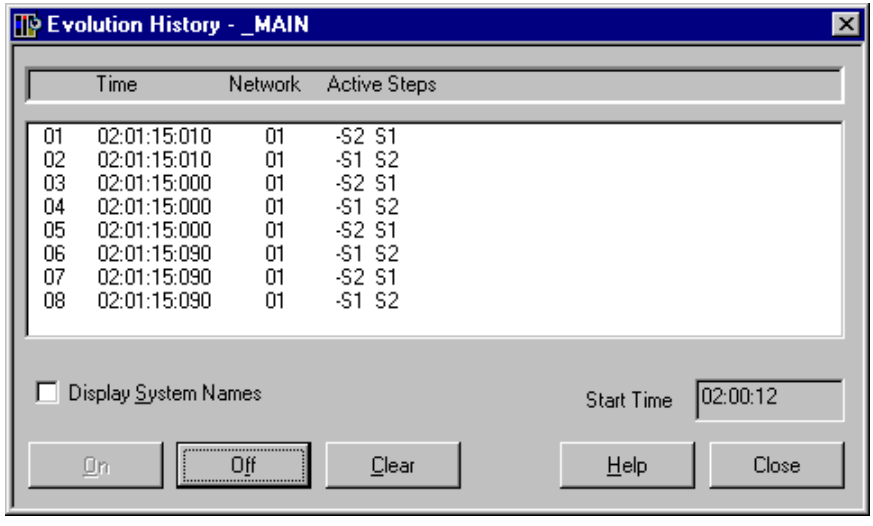
Recording and Viewing Evolution History

When the programmer is online and equal, you can record the evolution history of an SFC block. When this feature is enabled, a history of the last eight evolutions of the SFC block is recorded and displayed. The most recent evolution is displayed first on the list. Each evolution is time-stamped, and each newly active step for that evolution is recorded.

Once enabled, evolution history is continuously recorded in the PLC until the feature is turned off. To turn evolution history recording on or off, select the Debug menu and choose Evolution History. The Select Block dialog box will appear.



Use the cursor keys or mouse to select the block you want to record history for and click OK. The Evolution History window will open and display the message "Connecting to the PLC." To start recording Evolution History, click the On button.



The Evolution History window provides the following information:

Table

The evolution history display shows the evolution number (up to eight), Time generated, Network number, and Active Steps. The list of Active Steps shows when steps became active. (The steps are not necessarily currently active.)

Steps that have a time fault (active duration outside the time parameters) are indicated by a + (over) or - (under). For information on setting time parameters, see page 4-2.

Buttons

On

Enables the collection of evolution history of the current SFC block.

Off

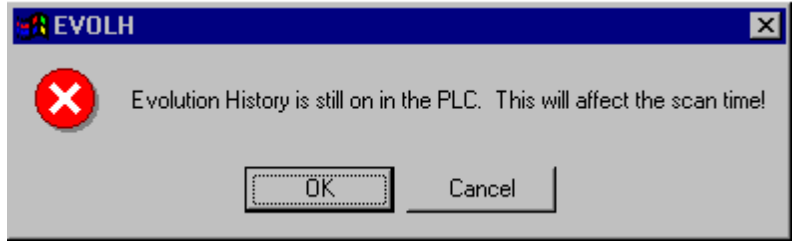
Stops evolution history recording.

Clear

Clears the evolution history display.

Close

Closes the Evolution History window. If you close the window while evolution history is still running, the following message will be displayed. To allow the evolution history to continue being collected, click OK. To turn evolution history collection off, click Cancel and then click the Off button. If you close the Evolution History window without turning it off, it will continue to run. The PLC's performance can be improved by turning evolution history recording off.



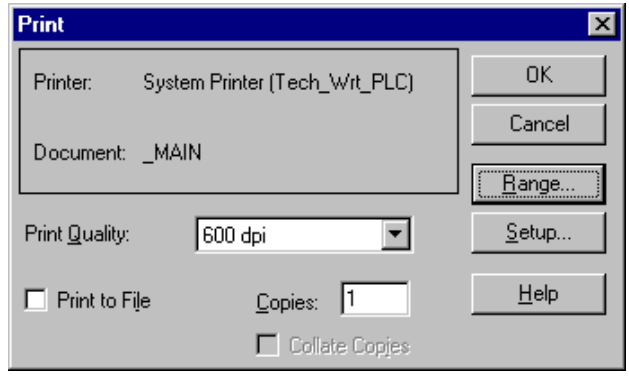
Start Time

Shows time when PLC history turned on.

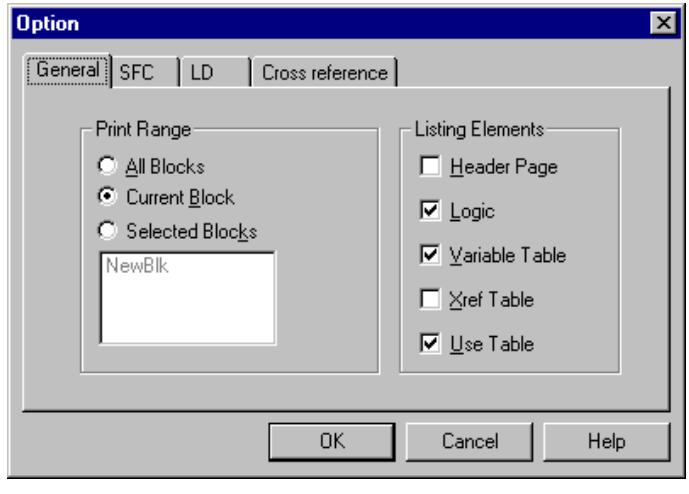
Chapter 5 Printing SFC Documents

To print your SFC program:

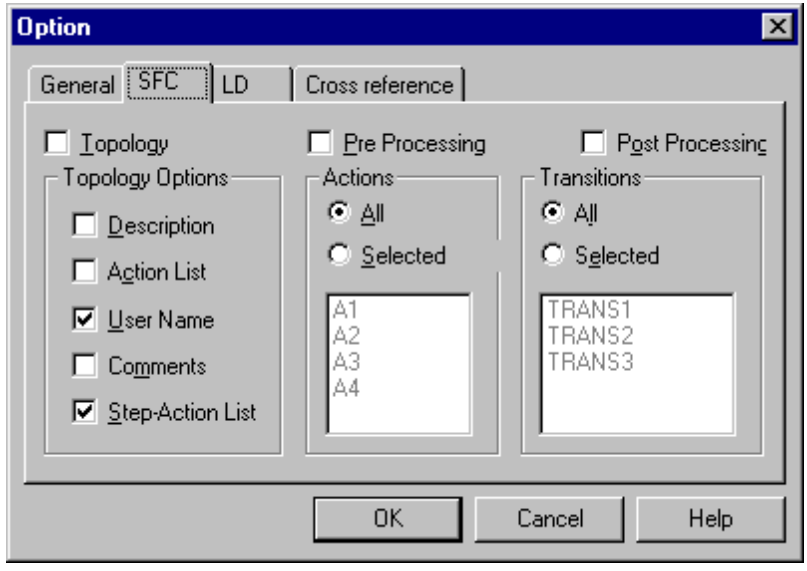
1. Select File from the SFC Editor menu and choose Print. The Print dialog box, which lists the active printer and allows you to customize print options, will appear.



2. To customize your program listing, click Range. The Option dialog box will appear.



3. To select the SFC topology details and the logic components to be printed, click on the SFC tab. (The LD and Cross reference tabs can be used to customize the terminal logic printout. For more information, see "Printing Your Program" in *Using CIMPLICITY Control*, GFK-1295.)



4. When you have finished setting the print options and selecting the SFC components to be included in the printout, click OK to start printing.

The SFC tab presents the following options:

Topology

Enables Topology Options for the selected SFC block.

Preprocessing

Prints preprocessing logic for the selected SFC block.

Postprocessing

Prints postprocessing logic for the selected SFC block.

Topology Options

Allows you to select the details that will be printed for each element. If none of the boxes in this group is checked, the SFC topology will be printed with system-generated names only.

Note

The Topology box must be checked for the Topology Options selections to take effect.

Description

Specifies whether descriptions associated with elements will be printed.

Action List

Specifies whether steps will be printed with a list of associated actions.

User Name

Specifies whether user-defined names will be printed. If this box is not checked, system-generated names will be printed. (This box is checked by default.)

Comments

Specifies whether comments associated with elements will be printed.

Step-Action List

Prints a list of all steps in the block and their associated actions.

Actions

Allows you to select printing of logic for all or specific actions. (If you click Selected and do not select any actions from the list, none will be printed.)

All

Specifies that the underlying logic of all actions defined in the SFC block will be printed.

Selected

Allows you to select actions for which the underlying logic will be printed. Actions defined in the block are shown in the list box.

Transitions

Allows you to select printing of logic for all or specific transitions. (If you click Selected and do not select any transitions from the list, none will be printed.)

All

Specifies that the underlying logic of all transitions defined in the SFC block will be printed.

Selected

Allows you to select transitions for which the underlying logic will be printed. Transitions defined in the block are shown in the list box.

Appendix A Common User Errors

This appendix describes error conditions that are unique to SFC logic blocks and to the ladder diagram logic found within them. It does not describe ladder diagram errors that are common to ladder diagram blocks and SFC blocks.

In general, anything that is an error in a relay ladder diagram logic block is also an error in the relay ladder diagram portion of an SFC block. The global checks that are made over an entire relay ladder diagram logic block include missing labels, backward MCR jumps, missing END_FORs, and too many program block calls. These global checks are also applied individually to each relay ladder diagram section of an SFC block. For example, every action logic section must have a matching label for each unique jump within that section of relay ladder diagram logic. The only exception is the check on the number of program block calls. For an SFC block, the check is made over the entire block, not the individual relay ladder diagram logic sections.

- General Errors A-2
- Transition Logic Errors A-2
- Transition Logic, Action Logic, and Preprocessing/Postprocessing Logic Errors A-3
- General SFC Element Errors A-3
- SFC Top Level Errors A-4

General Errors

The following errors will be detected while entering a relay ladder diagram rung:

- Reference to a transition variable name outside the relay ladder diagram transition logic for the variable.
- Attempting to write to a step.x or step.f flag.
- Attempting to write to a step.t value.

Transition Logic Errors

The following error conditions will be detected by the Analyze function. The block will remain *unexecutable* until these errors are corrected. (The block can be saved with the errors, but they must be corrected before the logic can be stored to the PLC.)

- Transition variable was not set.
- Transition logic does not contain exactly one regular (non-comment) rung.
- Transition logic sets variable(s) other than transition variable.

Other Logic Errors

The following error conditions in transition, action, or preprocessing/postprocessing logic will be detected when the program is compiled. The SFC block will remain unexecutable until these errors are corrected.

- A jump/label name is used in another relay ladder diagram logic section in this block.
- An MCR/END_MCR name is used in another relay ladder diagram logic section in this block.

The following error conditions *must* be corrected immediately:

- A backward MCR jump.
- Too many nested FOR loops.
- Too many EXIT_FOR loops.
- Attempting to flow more than 16 bytes of data between function blocks (instructions). You can use registers on the output of one function block and the input of the next function block instead.

General SFC Element Errors

The following errors will be detected at entry and must be corrected immediately:

- Attempting to add more steps, transitions, or unique connectors than the target PLC can handle. (See appendix B for target PLC limits.)
- Attempting to extend the SFC network beyond 255 rows or 255 columns.
- More than 32 parallel branches within one control structure.

SFC Top Level Errors

The following errors can occur at the main level of the SFC network. The block will remain unexecutable until they are corrected.

In addition to catching the errors that occur at the main level of the SFC, the Analyze function detects errors at all other levels of the SFC block. Errors are listed in the program Log window. The Analyze function starts automatically when the program is saved and can be started manually from the Program menu.

Detected immediately:

- Step and source connector directly connected.
- Multiple occurrences of the same destination connector.

Detected by the Analyze function:

- Transition not preceded by a step.
- Transition not followed by a step, source connector, or directed link.
- Disconnected SFC segments (two or more sections of an SFC network that are not connected by vertical links, horizontal branches, or connectors).
- More than one initial step.
- Missing initial step.
- Initial step occurs inside a simultaneous control structure.
- Destination connector that jumps into a simultaneous control structure from outside of it.
- Missing destination connector.
- Improper nesting of control structures. (Simultaneous convergences cannot be closed by selective convergences and vice versa.)
- Attempts to jump outside of a simultaneous divergence.
- Unreachable structures, that is, places in the topology that power flow will never get to.

- Illegal terminations in a simultaneous divergence. If a simultaneous divergence contains a selective divergence, and if any of the branches of the selective divergence reach a simultaneous convergence, all the branches of the selective divergence must reach the simultaneous convergence (none can end in a step).
- Convergences immediately followed by divergences with no elements in between.
- Illegal jumps. Connectors cannot be used to jump between branches in a simultaneous divergence or outside of the simultaneous divergence.
- Transitions cannot be immediately followed by more than one source connector.
- Transitions cannot be immediately followed by both steps and source connectors.
- Disconnected transitions. Transitions must be connected to elements on both sides.
- Connectors can have multiple sources, but only one destination.
- If a simultaneous divergence is within a loop (due to either a directed link or use of connectors), all branches within the simultaneous divergence must converge.

Appendix B PLC Limits

This Appendix contains a listing of SFC performance parameters for Series 90-30 and Series 90-70 PLCs.

The CIMPLICITY Control SFC Editor, release 2.0 supports the following PLCs.

Series 90-30 PLCs	Series 90-70 PLCs
CPU 361	CPU 731
CPU 360	CPU 771
CPU 352	CPU 772
CPU 351	CPU 781
	CPU 782
	CPM 914, 915
	CPM 924, 925

Table B-1. Target PLC Constraints for Series 90-70 PLCs

Feature	Release 6	Release 7	Release 8
Number of SFC networks per SFC block	1	1	1
Maximum number of steps per SFC network	255	255	255
Maximum number of steps per block	255	255	255
Maximum number of simultaneous active steps per SFC network	32	32	32
Maximum number of transitions per SFC network	383	383	383
Maximum number of transitions per block	383	383	383
Maximum number of action blocks per step	8	8	8
Maximum number of actions per block	511	511	511
Action Qualifiers	N: non-stored	N: non-stored	N: non-stored
Maximum use count per action	N/A	127	127
Maximum number of parallel branches in a simultaneous sequence per SFC network	32	32	32
Use of LD terminal language	yes	yes	yes
Maximum depth of evolution history queue	8	8	8
Number of blocks capable of being SFC blocks*	255	255	255
Maximum level of call nesting	8	8	8

*Constrained by memory

Table B-2. Target PLC Constraints for Series 90-30 PLCs

Feature	Release 5	Release 8
Number of SFC networks per SFC block	1	1
Maximum number of steps per SFC network	96	96
Maximum number of steps per block	96	96
Maximum number of simultaneous active steps per SFC network	32	32
Maximum number of transitions per SFC network	96	96
Maximum number of transitions per block	96	96
Maximum number of action blocks per step	1	8
Maximum number of actions per block	255	255
Action Qualifiers	N: non-stored	N: non-stored
Maximum number of parallel branches in a simultaneous sequence per SFC network	32	32
Use of LD terminal language	yes	yes
Macro-steps	no	no
Maximum depth of evolution history queue	8	8
Number of blocks capable of being SFC blocks*	1 (main)	65 (main plus 64 blocks)
Maximum level of call nesting	8	8

*Constrained by memory

Appendix C Glossary

Action count number	The number, at the upper right corner of a step, that indicates how many actions are associated with the step.
Against gravity	In the SFC Editor, this term indicates logic flow that goes from the bottom to the top of the topology. Logic that flows against gravity can be created using a directed link or connectors.
Cycle	A logical structure in which the flow of execution is transferred to a step that comes before the current step in the SFC network. This structure provides the ability to create a conditional loop. In the SFC Editor, the connector element is used to create this structure. (Also see jump.)
Directed link	A link that flows against gravity. A directed link must go from a transition to a step.
Gravity	In the SFC Editor, this term indicates that the normal logic flow goes from the top to the bottom of the topology and from left to right. Links between elements are created as gravity connections by default.
GUI	Graphical User Interface.
IEC	International Electrotechnical Commission.
Jump	A logical structure in which the flow of execution is transferred to a step that comes after the current step in the SFC network. This structure allows steps to be conditionally skipped. In the SFC Editor, the connector element is used to create this structure (also see cycle).
SFC	Sequential Function Chart programming language.
SFC element	An SFC element can be an initial step, step, transition, or connector.
SFC time fault	Occurs when a step's active time is less than the minimum time or exceeds the maximum time specified in the Set Time Parameters dialog box.
Terminal language	The underlying logic of the SFC program organizational units is written in another programming language. The terminal languages used in the SFC editor include Structured Text, Ladder Diagram, and Function Block Diagram. For protected mode target PLCs only, you can use more than one terminal language within an SFC network.
Terminal step	A step that is not followed by a transition.

A

Action
- logic, 2-3
- logic, editing, 3-34

Action count number
- defined, C-1

Action pool
- editing, 3-35

Action qualifiers, 3-38

Against gravity
- defined, C-1
- in basic control structures, 2-17

Analyze function
- errors detected by, A-4

Analyzing SFC networks, 4-2

Animation of Topology, 3-7

B

Basic control structures
- convergent selective sequence, 2-13
- convergent simultaneous sequence, 2-15
- cycle, 2-19
- divergent selective sequence, 2-12
- divergent simultaneous sequence, 2-14
- IEC compliance, 2-23
- jump, 2-18
- rules for, 2-20
- sequence loop, 2-17
- sequence skip, 2-16
- simple sequence, 2-11

Building an SFC network, 3-19

C

Comments
- adding to an element or action, 3-7
- indicated by *, 3-7

Confirm Action Replace, 3-32

Connector
- in a cycle structure, 2-19
- in a jump structure, 2-18
- using, 3-24

Convergence of simultaneous sequences, 2-21

Creating steps and transitions, 3-19

Cursor lock, 3-16

Customizing the Topology window display, 3-16

Cycle
- defined, C-1
- illustrated, 2-19

D

Deleting SFC elements, 3-33

Dialog boxes
- Action Pool, 3-21, 3-35
- Clear Pre/Post Processing, 3-40
- Confirm Action Replace, 3-32
- Connector Properties, 3-24
- Elements, 3-19
- GoTo, 3-26
- New Action, 3-14, 3-21
- New Transition, 3-15
- Pre-processing/Post-processing, 3-39
- Search, 3-26
- SFC Editor Options, 3-13
- SFC Limits, 3-18
- Step Properties, 3-33
- Step-Action Association, 3-37

Dialog boxes, diagnostics
- Add to Transition Monitor, 4-6
- Evolution History, 4-10
- Select Block (for evolution history, 4-10
- Set Time Parameters Warning, 4-3
- Transition Monitor window, 4-6

Dialog boxes, getting started

New Block, 3-3
New Main Block, 3-2
Dialog boxes, printing
Option, 5-1
SFC tab, 5-2
Directed link
creating, 3-24
example, 2-17
Display mode, 3-17
Documenting your SFC project, 3-40

E

Editing an SFC network, 3-26
Editing mode, 3-20
Editor options setting, 3-13
Elements
comments, 3-40
connecting, 3-23
copying, 3-31
cutting, 3-30
deleting, 3-33
disconnecting, 3-30
moving, 3-30
pasting, 3-31
properties, 3-33
searching, 3-26
selecting, 3-30
Errors, common user
general errors, A-2
general SFC element errors, A-3
logic errors, A-3
SFC top-level errors, A-4
transition logic errors, A-2
Evolution
SFC, 2-8
Evolution history
recording and viewing, 4-10
Examples
basic control structures, 2-11
convergence of simultaneous sequences, 2-21
nesting of dissimilar sequences, 2-22
non-IEC-compliant convergence, 2-23
non-IEC-compliant divergence, 2-24
SFC control sequence, 2-5
SFC evolution, 2-9

G

General errors, A-2
Getting started, 3-2
GoTo, 3-26
Gravity
defined, C-1
in basic control structures, 2-11
Grid display options, 3-16

H

Hot keys, 3-10
How elements are displayed, 3-6

I

IEC
defined, C-1
Importing a Logicmaster 90 SFC program, 3-3
Initial step, 2-3
Insert mode, 3-20
Insert network, 3-18
Installing the software, 1-2
Introduction, 1-1

J

Jump
defined, C-1
illustrated, 2-18

K

Key functions, 3-10

L

Limits, SFC
  Series 90-30, B-3
  Series 90-70, B-2
  viewing, 3-18
Lock cursor, 3-16
Logic errors, A-3
Logicmaster 90
  importing an SFC program, 3-3
  SFC Editor comparison, 3-42

M

Main SFC block
  creating, 3-2
Monitoring transition conditions, 4-6

N

Navigating in the SFC network, 3-26
Nesting of dissimilar sequences, 2-22
New SFC block
  creating, 3-3
Non-IEC-compliant convergence, 2-23
Non-IEC-compliant divergence, 2-24

O

Online documentation, 1-2
Output Log window
  analyze results, 4-2
  search results, 3-29
Override transition, 4-9
Overwrite mode, 3-20

P

Paste (with Logic), 3-32
PLC limits
  viewing, 3-18
PLC scan cycle, 2-6
PLCs
  supported by SFC Editor, 1-1, B-1
Pop-up menu, SFC, 3-9
Preprocessing/postprocessing logic
  clearing, 3-40
  editing, 3-39
Properties, 3-33

R

Recording and viewing evolution history, 4-10
Regular step, 2-3
Replace, 3-26
Reserved words, 3-2
Reset function, 2-25
RLD and SFC, 2-1
Rules for connecting elements, 3-25

S

Scan cycle, PLC, 2-6
Search and Replace, 3-26
Sequential function chart
  general SFC element errors, A-3
  SFC top-level errors, A-4
Setting time parameters, 4-2
SFC
  basic control structures, 2-11
  errors, A-1
  example sequence, 2-5
  IEC compliance, 2-23
  steps, 2-3
  transition, 2-4
SFC Editor
  options, setting, 3-13
  status bar, 3-8
  Terminal Language window, 3-12
  toolbar, 3-8
  Topology window, 3-6

Variable Declaration Editor window, 3-12
windows, 3-4
SFC elements
comments, 3-40
connecting, 3-23
copying, 3-31
creating, 3-19
cutting, 3-30
deleting, 3-33
disconnecting, 3-30
moving, 3-30
pasting, 3-31
properties, 3-33
searching, 3-26
selecting, 3-30
SFC evolution, 2-8
SFC limits
Series 90-30, B-3
Series 90-70, B-2
viewing, 3-18
SFC networks
analyzing, 4-2
creating, 3-18
SFC pop-up menu, 3-9
SFC_RESET, 2-25
Software
installing the software, 1-2
Status Bar
SFC, 3-8
Step status values
defined, 2-8
resetting, 2-25
Step-action associations
editing, 3-37
number of actions associated with step, 3-7
Steps
action logic, 2-3
initial step, 2-3
regular step, 2-3
setting time parameters for, 4-2
Summary information, 3-41
System-generated names
displayed, 3-6
in Variable Declaration editor, 3-12

T

Terminal language
defined, C-1
Terminal Language window, 3-12
Terminal step
defined, C-1
Time fault
defined, C-1
Time parameters
setting, 4-2
Toggle transition, 4-9
Toolbar, SFC, 3-8
Topology window, 3-6
Topology, animation, 3-7
Transition logic errors, A-2
Transitions, 2-4
editing logic, 3-34
forcing, 4-9
monitoring, 4-6
Troubleshooting
general errors, A-2
general SFC element errors, A-3
logic errors, A-3
SFC top-level errors, A-4
transition logic errors, A-2

U

Undoing a delete, 3-33
User-defined names
displayed, 3-6
Using online documentation, 1-2

V

Variable Declaration Editor window, SFC, 3-12
Variable names
system-generated, 3-12
user-defined, 3-6

W

What you will need, 1-2
Word-for-word changes, 3-41

Z

Zoom, 3-16